1000 Audio Modification

# Amazing COMPUTING™

Your Original AMIGA™ Monthly Resource

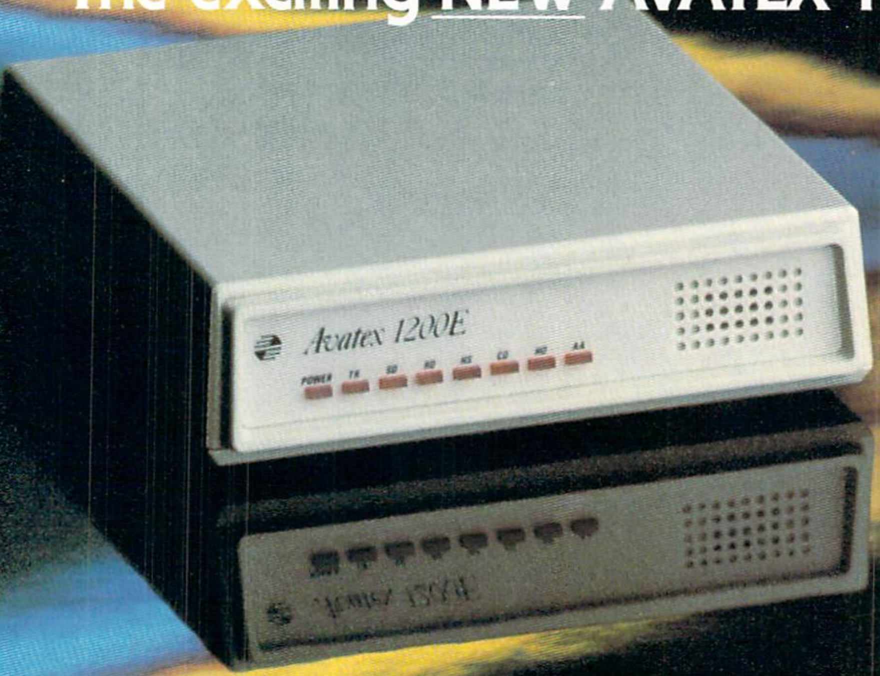
Volume 3 Number 3
US $3.50 Canada $4.50

## HARDWARE & COMMODORE AMIGA™

- PAL Help
- FACC II
- MIDI Compatibility for Your 2000

## Intro to Batch Processing

# GOOD THINGS COME IN SMALLER PACKAGES.
## The exciting NEW AVATEX 1200 E.



**$85**
**TOTAL DELIVERED PRICE.**

**AVATEX**
**Authorized Sales & Service**

**ACTUAL SIZE ONLY 5" X 6"**

**FREE! Communication Software & Compuserve Access Time with each MODEM.**

We give you a great price . . .
- + TOLL FREE ASSISTANCE
- + IMMEDIATE DELIVERY
- + NO CREDIT CARD SURCHARGE
- + NO SERVICE OR HANDLING FEES

+ **FREE SHIPPING!**

TOTAL HAYES COMPATIBILITY
8 LED STATUS INDICATORS
AUTO DIAL AND ANSWER
TONE OR PULSE DIALING

**NEW 1200 E FEATURES**
CCITT COMPATIBILITY
CALL PROGRESS DETECTION
INTERNAL SPEAKER
2 YEAR WARRANTY

Thousands of customers have purchased the AVATEX 1200 from Megatronics. New technologies now make it possible to offer an improved version of this popular Modem.

The 1200 E is amazing.
It weighs less and is smaller.
It has even more features.
It's still the same low price.

**MEGATRONICS**

MEGATRONICS, INC. P.O. BOX 3660, LOGAN, UT 84321

**To Order**
CALL FOR FREE CATALOGUE
CREDIT CARDS VERIFIED FOR YOUR PROTECTION

**800-232-6342**
INSIDE UTAH (801) 752-2642 FAX (801) 752-8752

We'll beat any advertised price.
MEGATRONICS, INC., P.O. BOX 3660, LOGAN, UTAH 84321

# LIVE!™

## AT LAST!

## ...real-time, LIVE! video on your Amiga's screen.

- ***True Color:*** just as it comes from your video source: camera, VCR, TV, anything. Direct, moving, in your Amiga's memory...our patented technology.
- ***Fast:*** video images in black & white, 32-color, and 4,096-color HAM. See 15 new images every second in black & white, 12 in color, 4 in HAM.
- ***Save:*** moving video, play it back, use it in other programs. Unlimited stills, too.
- ***Video Effects:*** real-time mouse-controled...posterization, fades, color-keying, strobes, more.
- ***Roll Your Own:*** programmer's video library, hardware documentation, examples in C, Basic.
- ***$295.*** Immediate delivery. This is hot.

To order call toll-free anytime:
Nationwide: 800-452-4445, ext. 1156
California: 800-626-9541, ext. 1156

*For more information, contact:*

$a^2$

A-Squared Distributions Inc.
6114 La Salle Avenue, Suite 326
Oakland, California 94611
415-339-0339

Amazing COMPUTING™

*Volume 3, Number 3*

# CONTENTS

## Amazing Features

**Take Five!** by Steve Hull 8
Blast your frustrations in these five new Amiga games.

**Desktop Video, Part IV** by Larry White 31
Put all the pieces together—the desktop video commercial.

**The Hidden Power of CLI Batch File Processing**
by Joe Rothman 35
Make your Amiga easier to use with CLI Batch files.

**A Conference with Eric Graham**
edited by John Foust 38
The mastermind behind Sculpt 3D and Animate 3D.

**Perry Kivolowitz Interview** by Ed Bercovitz 41
Amiga insights from a major developer and personality.

**Jean "Moebius" Giraud**
Interview by Edward L. Fadigan 55
Avant-garde art comes to the Amiga—in dazzling form.

**PAL Help** by Perry Kivolowitz 58
All the help you need for A1000 expansion reliability.

**Boolean Function Minimization**
by Stephen M. Hart 60
A useful digital design tool in AmigaBASIC.

**Amiga Serial Port and Midi Compatibility for Your A2000!**
by Lynn Ritter and Gary Rentz 68
Add an A1000-style serial port to your A2000!

**Electric Network Solutions the Matrix Way**
by Robert Ellis 70
Engineers! Practical routines for using matrix algebra.

**In The Public Domain** by C.W. Flatte 97
Hot flashes and highlights from the latest Fred Fish disks.

**The A.M.U.G. BBS List compiled**
by Joe Rothman, Chet Solace, and Dorothy Dean 100
514 BBS phone numbers in the U.S. and Canada.

## Amazing Reviews

**FACC II** by Graham Kinsey 16
Put a firecracker under your floppy drives.

**Uninvited** by Kenneth E. Schaefer 20
When was the last time a game scared you to death?

**Flow** by Pamela Rothman 25
Turn your brainstorms into mental works of art.

**Benchmark Modula-2 Compiler**
by Richie Bielak 81
Program development that beats Pascal to the punch.

## Amazing Columns

**Bug Bytes** by John Steiner 29
Stay abreast of today's bugs and tomorrow's upgrades.

**Modula-2 Programming**
by Steve Faiwiszewski 50
The gameport device and simple sprites in action.

**AmigaNotes** by Richard Rae 63
A1000! Create a software-switchable output filter.

**Roomers** by The Bandito 87
Inside AmiExpo ... Kickstart 1.4 ... Commodores 'R' Us?

**The Big Picture** by Warren Ring 89
Unwieldy arguments to system calls?
Discover the Unified Field Theory!

## Amazing Departments

Amazing Mail 4
Index of Advertisers 104
Public Domain Software Catalog 105

35MM SLIDES FROM YOUR ARTWORK!

AMIGA Specialists
TRU-IMAGE™

Professional 35mm Slides

- Now you can have reproduction and presentation quality slides of your work
- Distortion-free--fills in raster lines crisp bright colors, converts all IFF files

Now

Custom graphic art and illustration.

$10 each for your 1st to 4th slides.
5 to 9 slides-$8.50
Over 10 slides-$8.00
Add $2.00 for shipping.
New York residents add sales tax.
Call (212) 777-7609 FOR DETAILS
Ask for Ilene--or write TRU-IMAGE
P.O. Box 660, Cooper Station
New York, N.Y. 10276

Amiga is a trademark of Commodore-Amiga.

Amazing COMPUTING

ADMINISTRATION

| | |
|---|---|
| **Publisher:** | Joyce Hicks |
| **Assistant Publisher:** | Robert J. Hicks |
| **Circulation Manager:** | Doris Gamble |
| **Asst. Circulation Mgr.:** | Traci Desmarais |
| **Corporate Trainer:** | Virginia Terry Hicks |
| **Traffic Manager:** | Robert Gamble |

EDITORIAL

| | |
|---|---|
| **Managing Editor:** | Don Hicks |
| **Co-Editor:** | Ernest P.Viveiros Jr. |
| **Co-Editor:** | Michael T. Cabral |
| **Hardware Editor:** | Ernest P. Viveiros Sr. |
| **Music & Sound Editor:** | Richard Rae |
| **Amicus & PDS Editor:** | John Foust |
| **Copy Editors:** | Julie Landry<br>Michael Creeden |

PRODUCTION

| | |
|---|---|
| **Art Director:** | Keith Conforti |
| **Illustrator:** | Brian Fox |
| **Production Manager:** | Mark Thibault |
| **Associate Prod. Mgr:** | Rico A. Conforti |
| **Production Assistant:** | Keven V. Desmarais |

ADVERTISING SALES

**Advertising Manager:** John D. Fastino
1-800-345-3360 or
1-617-678-4200

SPECIAL THANKS TO:
Buddy Terrell & Byrd Press
Betsy Piper at Tech Plus
Bob at Riverside Art, Ltd.

Amazing Computing™ (ISSN 0886-9480) is published by PiM Publications, Inc., P.O. Box 869, Fall River, MA 02722. Subscriptions: in the U.S.12 issues for $24.00; in Canada & Mexico, $30.00; Overseas; $35.00. Printed in the U.S.A.

Copyright© 1988 by PiM Publications, Inc. All rights reserved.

First Class or Air Mail rates available upon request.

PiM Publications, Inc. maintains the right to refuse any advertising.

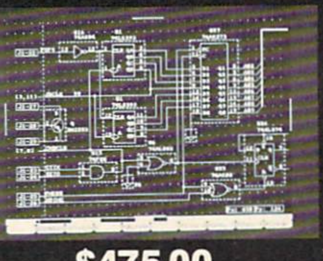
PiM Publications, Inc. is not obligated to return unsolicited materials. All materials requesting return must be received with a Self Addressed Stamped Mailer.

Send article submissions in both manuscript and disk format to the Co-Editor. Requests for Author's Guides should be directed to the address listed above.

Prolific Inc.

Available Now!

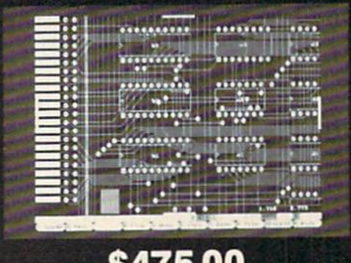
— PRO-NET —



$475.00

For AMIGA™ only

— PRO-BOARD —



$475.00

POWER+SIMPLICITY=PRODUCTIVITY

*Feel our POWER. . .*

PRO-NET:
- *Variable template size*
- *A to E paper size*
- *Extensive Library included*
- *Auto device number with Zone control*
- *Gate swapping*
- *Single click change to negative logic*
- *Innovative weight assignment*
- *Auto page reference*
- *Dynamic error checking*
- *Creates BOM, Spare Part List, Net List, Error report, etc.*
- *Supports printers & plotters*
- *Supports Laser Printer*
- *Back annotation from PCB layout*
- *Move, Rubber. . .and lots more*

PRO-BOARD:
- *.025 inch grid*
- *.001 inch grid Library*
- *12 mil trace, 13 mil space*
- *Produce 1, 2, 4 layer PCB*
- *Provides silk screen*
- *Auto coordinates assignment*
- *Auto produces power and grid layers*
- *Single line auto route*
- *Optional Net List Input for guided route, no need to look at schematic*
- *Prioritized route*
- *Dynamic error checking*
- *Supports printers, plotters and Gerber photo plotters*
- *Copy, Repeat. . .and lots more*

*See our SIMPLICITY. . .*

Intelligent Function Keys make our programs extremely user friendly, provide maximum screen area, always display all relevant commands, avoids excessive cursor movement and screen flashing between menu & drawing, guides user through operation, minimizes training time.

ORDER OR CALL FOR DETAILS
1808 W. Southgate Ave., Fullerton, CA 92633
Tel: (714) 447-8792 Telex: 5106016526 PROLIFIC CALIF
Western Union Easy Link Mail Box 62935949

TRY OUR DEMO DISKS FOR $15 EACH.

NEW!

*Also from Prolific Inc.,*

4 full feature AMIGA™ Macro Cross Assemblers for Z80, 6809, 8085 & 8051.

*See our SIMPLICITY*

— PRO-ASM —

*Feel our POWER*
- *Includes multi-pass Assembler, Linker, and Serial Down Load*
- *Generates relocatable Object Code Module*
- *Nested Macro*
- *Includes Files*

*Feel our POWER*
- *Conditional Assembling*
- *Rich set of directives*
- *Global and External Variables*
- *Data format includes Binary, Motorola Hex, Intel Hex, and Tek Hex*

$85.00

*DEALER INQUIRY INVITED.*

*AMIGA™ trade mark of Commodore Inc.*

# *Amazing Mail:*

Special: All letters with special tips and suggestions which appear with a star (*) will receive a coupon for five free PDS disks as a special thank you. If you have a special tip, please send your idea today.

**Dear Amazing Computing, ***
I have an Amiga 500 and an Okimate 20 color printer with an Amiga 1000 Plug 'n Print module. I have found that this cable will not work in the A500. Here is a cable that works fine for me:

| Okidata Amphenol 36 | Amiga 500 DB 25 | call or function |
|---|---|---|
| 1 | 1 | strobe |
| 2 | 2 | D0 (LSB) [DATA] |
| 3 | 3 | D1 |
| 4 | 4 | D2 |
| 5 | 5 | D3 |
| 6 | 6 | D4 |
| 7 | 7 | D5 |
| 8 | 8 | D6 |
| 9 | 9 | D7 |
| 10 | 10 | acknowledge |
| 11 | 11 | busy |
| 12 | 12 | paper out |
| 13 | 13 | select |
| 14 | 17 | ground |
| 31 | 14 | +5 ( I PRIME) |

Perhaps this simple cable will save someone a lot of frustration.

Thank you,
Gary B. Lee
New York

*Thank you for the tip.*

**Dear Amazing Computing, ***
After receiving your latest issue V3.1, I typed in the "BSpread" program, and made some considerable improvements to it.

I added a "raise to the power of" operation in addition to the basic "+-*/" operations.

I improved the YesNo routine, so it would recognize Keyboard input of "Y/N" for Yes or No, as well as clicking on the appropriate gadget.

I let the arrow keys control Cell Selection, as well as keeping the ability to select a cell with the mouse pointer.

And finally, I increased the number of menu items with keyboard alternates. Speaking of which, an undocumented feature of Catley's program is, when in the Cell Selection Mode, typing a "$" will select "Zero Cell" - Menu Item.

Unfortunately, with my extra routines, and only 512K memory, I had to delete the "Remove Row" command. Also, I have a subroutine to read the initial Color Register values, so that the program will know what to reset the palette values to on exit. But again, I have insufficient memory for those routines.

Sincerely,
Jonas S. Green
Massachusetts

*Thank you for extending this program as we hoped all our readers will. Programs are published in AC as starting points, not completed projects. Each suggestion is presented to spur the reader on to a better understanding of the Amiga. By extending a published program and letting us know, you have paid AC and Mr. Catley a very high honor. If you send a disk with these refinements, we will add the code to a future Public Domain Disk and send you a copy as a further thank you.*

**Dear Amazing Computing, ***
How careless of you to overlook the expiration of my subscription! In an age where I am accustomed to getting "hurry, renew now" notices from lesser publications before the current subscription check has even cleared the bank, it's refreshing not to be bombarded with junk mail for months before the event. If some of these publications put half as much effort into editorial content as they do on promoting subscriptions, I might be inclined to resubscribe. On the other hand, I think AC is doing a great job of supporting the Amiga community, and I happily enclose my check for another year.

Unlike the glossier Amiga specific publications, I think AC has grown with the Amiga and its user base, and I'd like to encourage you to continue to do so. We're not all beginners at this anymore, and though articles to help the new user are certainly useful, you can only run so many 'tips' on how to update your startup sequence. I enjoy your more technical articles and programming tutorials and examples. I'd like to see more hardware projects, and would especially like to suggest a follow up article to your original 5 1/4" drive interface project. Although that circuit was fine for Transformer, it lacks the ID logic circuitry required to get it to operate as a half-size Amiga disk under 1.2, or to be recognized by the kernel to run PCUtils. I'd also like to see more articles on programming the Amiga in assembler, with examples of code that goes beyond the "hello world" type of example. Explanations of how to find various system control blocks, how various Exec calls work, etc., would be interesting to a lot of people who have worked with low-level code on simpler machines and find the Amiga's environment a bit big and confusing.

I especially appreciate the new Bug Bytes column--it's often difficult to get this sort of information. I've enclosed a submission that might help another user avoid wasting a lot of time or a long distance phone call. Please continue this column and add as much of this type of information as you can get. It's a great service to the users of the software and a great way to judge responsiveness of different software publishers to the needs of their users before committing money to buying a product from them. By publishing this sort of column, you expose the purveyors of sloppy code and reward those who are willing to correct the inevitable bugs that do get through.

Once again, thanks for putting out the best Amiga magazine on the market.

# *Amazing Dealers*

The following are Amazing Dealers, dedicated to supporting the Commodore-Amiga™.
They carry Amazing Computing™, your resource for information on the Amiga™.

If you are not an Amazing Dealer, but would like to become one, contact:

PiM Publications, Inc.
P.O.Box 869
Fall River, MA. 02722
1-617-678-4200

| | |
|---|---|
| **Alabama** | |
| Abax Data Systems | Huntsville |
| Alabama Book Store | Tuscaloosa |
| Command Computers | Birmingham |
| Electronics Boutique | Hoover |
| Mall Tel Communications | Mobile |
| Mel's Photo & Computer Shop | Montgomery |
| Universal Computer System | Mobile |
| **Arkansas** | |
| Computermart | Mountain Home |
| The Micro Shop | Little Rock |
| The 64 Store | Drytheville |
| The 64 Store | Jonesboro |
| The 64 Store | North Little Rock |
| **Arizona** | |
| Computer Libray, Inc. | Tempe |
| Computer West | Phoenix |
| Computer Works | Glendale |
| Copperstate Business Systems | Phoenix |
| Hacko's | Tempe |
| North American Digital | Tucson |
| **California** | |
| Adsy International | Foster City |
| Augnet | Fremont |
| Balboa Office Supply | San Diego |
| Bits n Pieces Comp Prod. | Anaheim |
| Brown Knows Computing | Redlands |
| Byte Factory Computers | Salinga |
| Century Computer Systems | La Habra |
| Circus of Books | Los Angeles |
| Circus of Books | Los Angeles |
| Commodore Land | Alhambra |
| Commotech | National City |
| Compunik | Los Angeles |
| Computer Attic | Palo Alto |
| Computer Cellar | Anahiem |
| Computer Nook, Inc. | San Bernadino |
| Computer Outlet | San Diego |
| Computer Outlet | San Marcus |
| Computer Time | Citrus Hts |
| ComputerLand Fremont | Fremont |
| Coronet Newsstand | Oceanside |
| Dataphile | Watsonville |
| Delta Spectrum | Laguna Beach |
| Discount Computer Station Inc. | Long Beach |
| Empire Computers | Santa Rosa |
| ECX Computer Co. | Walnut Creek |
| Harding Way News | Stockton |
| Heinrichson, Schneider & Young, Inc. | Berkeley |
| Home Computing Center | Concord |
| Home Computing Center | Mill Valley |
| Home Computing Center | San Leandro |
| Home Computing Center | San Bruno |
| HT Electronics | Sunnyvale |
| K J Computers | Granada Hills |
| Levity | North Hollywood |
| LTC Computers | Santa Ana |
| Micro Flight | San Jacinto |
| MTS Inc. Tower Books | Sacremento |
| Nonagon Software | El Cerrito |
| Op Amp Tech Books | Los Angeles |
| Paras Newsstand | San Diego |
| Put's Electronics | Rio Linda |
| Real Computers & Electronics | Petaluma |
| Ridgecrest Computer Ctr. Inc. | Ridgecrest |
| Sawyer's News Inc. | Santa Rosa |
| S.O.S. Computers | Los Angeles |
| Software First | Santa Rosa |
| Software Cellar | Anahiem |
| Software Central | Pasadena |
| Software Plus | Citrus Hieghts |
| The Booksmith | San Francisco |
| Tower Books | Chico |
| Tower Books | Concord |
| Tower Books | Mountain View |
| Tower Books | Sacremento |
| Tower Books | Sacremento |
| Tower Books | Sacremento |
| Tower Books | San Mateo |
| The Computer Room | Scotts Valley |
| Warner Engineering | San Diego |
| Winner's Circle | Berkley |
| World Book & News | Hoolywood |
| **Colorado** | |
| Aurora Newsstand | Aurora |
| Book Tech Distributing | Englewood |
| Citadel Computers | Colorado Springs |
| Computer Discount | Denver |
| Computer Room | Aurora |
| Ebert Personal Computers | Aurora |
| Micro Computer World | Fort Collins |
| Micro World | Lakewood |
| Sun Country Computers | Colorado Springs |
| **Connecticut** | |
| Aetna Life Club Store | Hartford |
| Amigo Business Computer | Norwalk |
| Computer Outlet | PlaVernon |
| Friendly Computers | Plantsville |
| Mnemonics Inc. | Norwalk |
| Personal Computer Center | Norwich |
| Software City | Orange |
| Software Kingdom | East Windsor |
| Spectrum Computers | New Haven |
| Tricom Computers | North Windham |
| **Delaware** | |
| Castle Video | Newark |
| **Florida** | |
| A A Computers | Jacksonville |
| Commodore Country Inc. | Pinellas Park |
| Computer Bar | Pennsacola |
| Computer Base | Brandon |
| Computer Image | Miami |
| Computer Specialty | Melbourne |
| Computer Terminal Inc. | Sebring |
| Computers 4 Rent | W. Palm Beach |
| Computers Plus | S. Daytona |
| Cotronics Inc. | Stewart |
| Data Base | Davie |
| Education Computers Etc. | Talahasse |
| Family Computers | Sarasota |
| Florida Book Store | Gainesville |
| Heartland Computers | Mulberry |
| Joe's News, Inc. | Miami |
| K&L Computers | E. Palatka |
| Megaport Computer Center | Venice |
| Micro Computer Center | Largo |
| New Age Electronics | St. Petersburg |
| Program Store | Clearwater |
| Random Access Computers Inc. | Ft. Walton Beach |
| Software City | Orange Park |
| Software Software | Plantation |
| Sunny's At Sunrise, Inc. | Sunrise |
| The Open Door | Cocoa |
| The PC Collection | Coral Gables |
| **Georgia** | |
| Future System | Augusta |
| Software South Inc. | Savannah |
| The 64 Store | Atlanta |
| **Hawaii** | |
| Astro Computers & Electronics | Aiela |
| Computer House Inc. | Honolulu |
| J & C Repair | Aiea |
| **Idaho** | |
| A.B.I. Computer & Video | Nampa |
| Boise Computer & Video Inc. | Boise |
| **Illinois** | |
| A Bitkeeter | Rockford |
| Computerland O'Hare | Park Ridge |
| ComputerLand of Niles | Niles |
| DigitalWorld | Addison |
| I.C.V. Data Center Inc. | Mundelein |
| Illini Microcomputers Inc. | Naperville |
| Kroch's & Brentano's | Chicago |
| Micro Age | Oaklawn |
| Software Or Systems | Fairview Hts. |
| Software Plus | Wheeling |
| Software Plus West | Hanover Park |
| The Memory Expansion | Bloomington |
| Unique Computer | Sterling |
| **Indiana** | |
| Burkat Computer Ctr. | South Bend |
| Bytrex of Angola | Angola |
| Computer Corner | Fort Wayne |
| Computer Corner | Marion |
| Computer People Inc. | Michigan City |
| Greg Chaney | Greenfield |
| Indiana News | Indianapolis |
| Micro Computer Centers Inc. | Indianapolis |
| Spectra Computer Sys. Inc. | Evansville |
| Vons Computers | West Lafayette |
| **Iowa** | |
| Gronet Computers | Des Moine |
| Micro Computer Applications LTD | Marshaltown |
| **Kansas** | |
| J & L Electronics | Liberal |
| MidKansas Computers | Newton |
| Midwest Computers | Manhattan |
| Software City | Overland Park |
| **Kentucky** | |
| Computer Dimensions Inc. | Lexington |
| Computer Store of Kentucky | Louisville |
| Hawley Cooke | Louisville |
| Reality World Computers | Lexington |
| **Louisiana** | |
| Computer Time | Shreveport |
| Modern Business Machines | Metairie |
| Software Center International | Metairie |
| Software Solutions | Baton Rouge |
| Software Source | Lafayette |
| The Bok Shelf | Slidell |
| The Computer Clinic | Lafayette |
| **Maryland** | |
| Buried Treasure | Rockville |
| Computerware TODAY | Gaithersburg |
| Compuvision Vision Center | Columbia |
| Layhill Newsstand | Kensington |
| Micro Computer Center | Baltimore |
| Software Advantage | Rockville |
| Software N Things | College Park |
| Waldorf Computer | Waldorf |
| **Massachusetts** | |
| Club Computer | Cambridge |
| D-Five | Bedford |
| DOX | Williamstown |
| General Computer Store | Framingham |
| HCS Computer Center | Marshfield |
| LCA Video & Computer Ctr. | Norwood |
| Memory Location | Wellsley |
| OmniTek Computers | Tewksbury |
| Pioneer Valley Data Equipment | Amherst |
| Softpro | Burlington |
| The Bit Bucket | West Newton |
| The Software Haus | Quincy |
| Tycom Inc. | Pittsfield |
| **Maine** | |
| Computer Barn | Damariscotta |
| Hands On Computing | Westbrook |
| Radio Shack Associate Stores | Jay |
| **Michigan** | |
| Basic Computer Center | Grand Rapids |
| Bits-Bytes-Nibbles | Petowskey |
| Canton Computer | Canton |
| CompuSoft | Kalamazoo |
| Direct Access | Lansing |
| Edmunton Computer Center | Pontiac |
| Global Computer Center | Dearborn |
| Central Park Computers | Okemos |
| Michigan Software | Novi |
| Midland Computer Shop | Midland |
| Pro-Video | Okemos |
| Programs Unlimited | Grand Rapids |
| Readmore Newscenter | Lansing |
| Roseville Computer Store | Warren |
| Slipped Disk | Madison Heights |
| Specialists Computer | Livomia |
| State Street Computers | Ann Arbor |
| T.J. Enterprises | Clinton |
| The Edge Connector | Clio |
| The Soft Warehouse | Kalamazoo |
| Ye Olde Computer Shoppe | Ypsilanti |
| **Minnesota** | |
| Computers, etc. | Eagan |
| JMH Software | Maple Grove |
| MCD of Hibbing Inc. | Hibbing |
| Midwest Computer Services | Clearwater |
| Shinder's Read More | Minneapolis |
| Specialists IN | Hopkins |
| Valiant Inc. | Stillwater |
| **Mississippi** | |
| National Computer Center | Ocean Springs |
| **Missouri** | |
| Associated Computer Services | Springfield |
| Brands Mart Computers | Kansas City |
| Data Byte Computer Co. | St. Louis |
| Instant Replay Ltd. | Chesterfield |
| Systems Plus | Independence |
| Systems Plus of St. Louis | St. Louis |
| **Montana** | |
| Appelgren Computer Systems | Great Falls |
| **Nebraska** | |
| BULLS-EYE | Lincoln |
| My Com Inc. | Plattsmouth |
| Double E Electronics | Omaha |
| **Nevada** | |
| Century 23 Inc. | Las Vegas |
| Computer World | Las Vegas |
| Tower Records | Las Vegas |
| **New Hampshire** | |
| Diversified Computers | Keene |
| **New Jersey** | |
| Family Computer Centres | South Orange |
| Morris County Stationers | Chester |
| **New Mexico** | |
| New Horizon Computer System | Alamogordo |
| Page One Newsstand | Albuquerque |
| Technological Concepts | Las Cruces |
| **New York** | |
| Amuse | New York |
| Amicus- Hudson Valley | Wappingers Falls |
| Amigo Business Computer | East North Port |
| Byte Shop | Merrick |
| Compurite Center | White Plains |
| Computer Outlet | Jamestown |
| Computer Palace | Feldon |
| Computers Etc. | Syracuse |
| Consumers Software | Commack |
| Cromland Inc. | Brooklyn |
| Leigh's Computers | New York |
| Mnemonics | North Port |
| N.U.B.S. Computer Center | Bronx |
| Softwaire Centre International | Forest Hills |
| Software & Such | Scotia |
| Software City | Forest Hills |
| Software Supermarket | Tonawanda |
| Star Tech Systems | Massena |
| Synfaire Trading | New York |
| Tadiran | New York |
| Tedrow Business Products | Rochester |
| The Computer Cellar | Albany |
| The Working Computer | Stoney Brook |
| Video Computer Bin | Brooklyn |
| Video Computer Center | Rome |
| Video Connection | Schenectady |
| Village Computer Inc. | New York |
| World Computers Inc. | Hicksville |
| **North Carolina** | |
| C2 of Hickory | Hickory |
| Digitz | Raleigh |
| K&S Newsstand | Clemmons |
| K&S Newsstand | Kernersvile |
| K&S Newsstand | Winston-Salem |
| K&S Newsstand | Winston-Salem |
| K&S Newsstand | Winston-Salem |
| Software City | Raleigh |
| Software City | Charlotte |
| TDS Computers | Durham |
| Triad Computers | Winston-Salem |
| **North Dakota** | |
| Computer Associates | Fargo |
| G. F. Computers | Grand Forks |
| The Computer Store | Minot |
| **Ohio** | |
| Computer Network | Newark |
| Computer Site | Cleveland |
| Earthrise Micro Systems | Columbus |
| Fairborn Home Computer | Dayton |
| Lakes Consumer Electronics | Akron |
| Main Office Supply | Coshocton |
| Micro Center | Columbus |
| Microwave Magic | Fairfield |
| North Coast Programming | Willoughby |
| Personal Computer Co. | Youngstown |
| Paper & Ink | Tiffin |
| Quality Computer Application | Toledo |
| Saxton Computer Center | Sandusky |
| Softwaire Centre International | North Olmsted |
| Software and More | Cincinnati |
| Software City | Centerville |
| Software Connection | Mayfield Hts. |
| **Oklahoma** | |
| Colonial Video & Computer | Bartlesville |
| Micro Link Computer Store | Tulsa |
| P C Tech | Tulsa |
| Second Hand Software | Oklahoma City |
| Video-Comp Inc. | Lawton |
| **Oregon** | |
| Andromeda | LaGrande |
| Capitol News Center | Salem |
| Clackamas Computers | Beaverton |
| Clackamas Computers | Clackamas |
| Computron Business Systems | Portland |
| Powell's Books | Portland |
| Powell's Tech | Portland |
| Rich Cigar Store | Portland |
| Software Express | Eugene |
| **Pennsylvania** | |
| Alpha Omega EBE | Warren |
| Basic Computer Systems | Hermitage |
| Chester County Book Co. | West Chester |
| Circle Computer | Ephrata |
| Downington Computer Center | Downington |
| East Coast Software | Hershey |
| Floppy Disk | Mechanicsburg |
| Gene's Books | King of Prussia |
| North Central Digital Systems | Danville |
| Pittsburgh Computer Store | Pittsburgh |
| Triangle Computers | Indiana |
| **Rhode Island** | |
| Micro Limits | Smithfield |
| Micro Limits | Warwick |
| Software Connection | Warwick |
| **South Carolina** | |
| Avcom International Inc. | West Columbia |
| Horizon | Greenville |
| Micro Computer Depot | Sumter |
| Software City | Spartanburg |
| Watson's Jewlery | Chesterfield |
| **South Dakota** | |
| Computer Business Center | Rapid City |
| **Tennessee** | |
| ComputerLab of Memphis | Memphis |
| The 64 Store | Bartlett |
| **Texas** | |
| Bookstop | Houston |
| Byte Shop | Austin |
| C.E.M. Corporation | Brazoria |
| Colonel Video | Houston |
| Computer Age | Houston |
| Computer Age | Dallas |
| Computer Magic | Austin |
| FNS Computer Services | Texarkana |
| Metropolitan Computer | Richardson |
| Micro Search | Houston |
| Software Library | Witchita Falls |
| Software Terminal | Fort Worth |
| The Computer Experience | San Antonio |
| The Computer Stop | Abilene |
| **Utah** | |
| Computers Plus | Murray |
| Magazines and Gifts | Salt Lake City |
| People's Computers | Provo |
| **Vermont** | |
| Ray Supply Inc. | South Burlington |
| **Virginia** | |
| Alonso | Alexandria |
| Diskovery | Falls Church |
| Family Computer Center | Fairfax |
| I/O Computer | Norfolk |
| Software Central | Norfolk |
| Software City | Richmond |
| Virginia Micro Systems | Woodbridge |
| **Washington** | |
| A.P.P.L.E. CO-OP | Renton |
| Apple Valley Computers | E. Wenatchee |
| Bellingham Computer Center | Bellingham |
| Bits, Bytes and Nibbles | Spokane |
| Bull Dog News | Seattle |
| Byte World | Kent |
| Computer Exchange Services | Bellingham |
| Computer Quest | Edmonds |
| Computers + Inc. | Tacoma |
| Computers and Movie Magic | Yakima |
| Family Computers | Lynwood |
| KAOS | Northeast Poulsbo |
| Nybbles & Bytes | Tacoma |
| Omni International | Seattle |
| Programs Plus | Seattle |
| Real Time Systems | Seattle |
| Take A Byte | Kennewick |
| **Wisconsin** | |
| Colortron Computers | Racine |
| Computer Software Center | Milwaukee |
| Fox Valley Personal Computer | Oshkosh |
| Mom's Computers | Stevens Point |
| Richard Reale | Madison |
| TMW Software Inc. | Wausau |
| **Wyoming** | |
| Byte Emporium | Laramie |
| **Canada** | |
| Compute or Play | Alberta |
| Computerworks | Alberta |
| Computer Shop | Alberta |
| Data Pro | Alberta |
| Computer Shop of Calgary | Alberta |
| The Software House | Alberta |
| Hindsin Computing | Alberta |
| Red Deer Micro Systems | Alberta |
| Memory Land | Alberta |
| Computer King | British Columbia |
| Conti | British Columbia |
| Nutec | British Columbia |
| Sprita | British Columbia |
| Strider | British Columbia |
| Super Software | British Columbia |
| Canadian Computer Era | Manitoba |
| Micromart | Manitoba |
| ComputerWorld | Manitoba |
| J.W. Allen | Manitoba |
| Kobetak | Manitoba |
| Minicomp | Manitoba |
| Arkon | Ontario |
| Chatam Computer | Ontario |
| Computerland | Ontario |
| Comspec | Ontario |
| Electronics 2001 | Ontario |
| Hastings Data | Ontario |
| The Computer & You | Ontario |
| Thornhill | Ontario |
| Software Co. | Quebec |
| Deltronie Solare | Quebec |
| Mierodata | Quebec |
| Info Mereier | Quebec |
| Laval Micro | Quebec |
| Vision Tronique | Quebec |
| Distribution Visipro | Quebec |
| **Distributed in Canada by** | |
| Ingram Software Distributors | Concord, Ontario |
| Compulit Distributors | British Columbia |
| Phase 4 Dist. Inc. | Calgary, Alberta |
| **Argentina** | |
| Logic Computer | Buenos Aires |
| **Australia** | |
| Computermart | Leederville,WA |
| Diskworks | Greenwich, NSW |
| High Technology Computer Supplies | Brighton, Melborn |
| John Verhoven | Cloverdale, WA |
| Maxwell Office Equipment | Abbotsford, Melburn |
| Megadisk | Lane Cove |
| Megatronics | Lanecove,NSW |
| **Saudia Arabia** | |
| Abdulla Fouad & Sons | Daman |
| **Scotland** | |
| The Amiga Center of Scotland | Edinborough |
| **Sweden** | |
| Nicholas Peerson | Lund |
| **Switzerland** | |
| Software Land AG | Zuerich |
| **West Germany** | |
| Ralph Hollax | Huerth |

Amazing Computing™ is also available in most B. Dalton Booksellers, B. Dalton Software Stores, Crown Books, and Software Etc. locations.

Please continue to devote your first efforts to quality content and leave the color glossies at the bottom of your priority list. I'm very pleased with the magazine and I feel you have been a great to the Amiga community and to the success of the Amiga.

**Re: "Unable to load Instrument from Disk" message from Aegis SONIX.**
A common problem encountered by users of Aegis SONIX (2.0) is an error message "unable to load Instrument from Disk" that occurs when trying to load an individual instrument from the directory list or when trying to load a score that requires instruments that appear in the current Instruments directory. The problem becomes especially apparent if you try to run SONIX on a single-drive system.

The underlying problem is that each of the 3 types of instruments has a corresponding ".tech" file containing code, presumably to handle that type of instrument. Since SONIX disables disk requestors, if you attempt to load a type of instrument that has not been loaded before while the program disk is not available, you get an error message without clear explanation of what the problem is, or an opportunity to swap in the program disk.

To circumvent this problem, keep at least one instrument of each type on the program disk, and load one of each at the beginning of your SONIX session. Once you have done this, the files are loaded and are available for additional instruments of the same type.

Other than this problem, I've had no trouble at all running SONIX with a single drive, though it is documented to require two drives.

Sincerely,
Scott S. Denham
Texas

*Thank you for the high praise and the good ideas. AC is always open to suggestions for improvement and your enclosed suggestions will be placed high on our lists.*

**Dear Mr. Hicks: ***
In previous letters that I have addressed to various members of your editorial staff (Mr. Viveiros, Mr. Cabral), I have pointed out grammatical errors, in *Amazing Computing*, that reveal ignorance of, and disregard for, the correct usage of certain Latin-derived words, especially the words "data" and "media," much used today and usually misused, in your journal and elsewhere.

The recent issue of Amazing Computing (Vol 3, No 2, page 39, third paragraph) contains the latest blooper that I cannot allow to go unchallenged:

"This data is then processed ......" should, of course, be "These data are then processed......".

Doesn't anyone proof-read articles before they are sent to the printer? Doesn't anyone on the editorial staff care about the level of literacy displayed by your publication?

I am getting the impression that your staff is insensitive regarding language and indifferent to the quality of the prose that you pass into print. I also have the impression that you people disregard my appeals for improved literary standards. When you display indifference, you send that signal to readers. Your last editorial spoke of the support provided by readers. My letters, in addition to my subscription fee, are my way of trying to support your endeavor--by trying to elevate, just a little, the level of literacy of your magazine and mine.

William Dzombak
Pennsylvania

*Thank you for your concern and obvious interest. I wish to assure you Amazing Computing has steadily increased its copy editing diligence, yet no matter how often we read or reread an article, errors can and do creep into print. The AC staff has been increased to handle the demand of producing the Amiga monthly magazine with the largest editorial content. Our best efforts to produce smooth, understandable text clash when faced with words and phrases exclusive to the Amiga. This jargon sometimes appears odd and even incorrect when compared to appropriate English formats. We are making every effort at retraining and restraining these spirited new people.*

*However, as to the use of the word "data." As defined by Webster's New Collegiate Dictionary © 1976 by G. & C. Merriam Co., "Data, a noun plural, but singular or plural in construction [plural of datum]." Which means "data" is plural, yet the writer has the option of using either a singular or plural verb.*

*Common usage and perception of data (both word and concept) make the singular use the preferred use. Data, as a plural, sounds awkward to the ear and tends to distract the reader from the material. Since communication is our goal, distracting the reader is counterproductive.*

*If this sounds as if the American/English language is deteriorating through use, please remember our language's greatest strength is in its expanding use. Our language takes the best from other languages and incorporates these ideas and expressions into its expanding vocabulary. Just like the Amiga.*

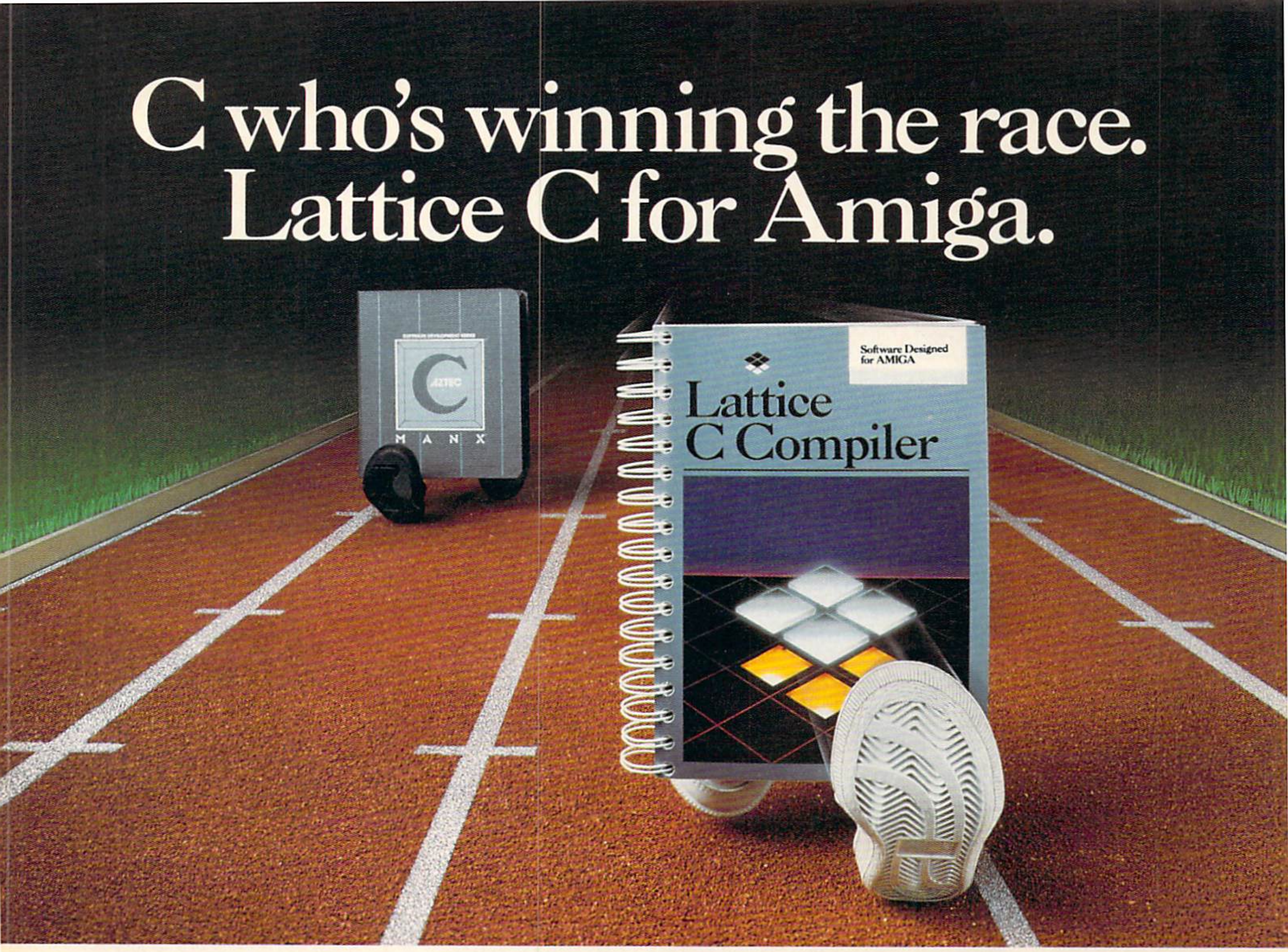
*Our task at Amazing Computing is to present a constant stream of interesting, topical, pertinent, and valuable information for the entire Amiga community. It has been our goal from the beginning to provide this material through Amiga users who wish to share their observations, successes, and delights in their Amigas.*

*We encourage our authors to produce articles based on interest--theirs and our readers. We continue to stress content over style, the concept: grammar can be corrected, but valuable information cannot be manufactured and should not be ignored.*

*Since our first priority has been and always will be to provide this style of information, we are at a disadvantage to publications of a more literary and artistic nature. But, we don't mind.*

•AC•

# C who's winning the race. Lattice C for Amiga.



Lattice C has long been recognized as the best C compiler. And now our new version 4.0 for Amiga™ increases our lead past the competition even further.

**Ready, set, go.** The new Lattice AmigaDOS C Compiler gives you faster, more efficient code generation and support for 16 or 32-bit integers. There's direct, in-line interface to all Amiga ROM functions with parameters passed in registers. What's more, the assembler is fully compatible with Amiga assembler syntax.

**More great strides.** The linker, Blink, has been significantly enhanced and provides true overlay support and interactive recovery from undefined symbols. And you'll have a faster compile and link cycle with support for pre-linking.

**There's no contest.** Standard benchmark studies show Lattice to be the superior C language development environment. With stats like these, it's no wonder that Commodore-Amiga has selected Lattice C as the official Amiga development language.

| | Lattice® Version 4.0 | Manx® Version 3.40 |
|---|---|---|
| **Dhrystone** | 1294 Dhrystones/second | 1010 Dhrystones/second |
| **Float** | 22.20 Secs. (IEEE Format)<br>10.16 Secs. (FFP Format) | 98.85 Secs. (IEEE Format)<br>17.60 Secs. (FFP Format) |
| **Savage (IEEE)** | 47.67 Secs./.000000318 Accuracy | 119.6 Secs./.000109 Accuracy |

**Going the distance.** You'll experience unsurpassed power and flexibility when you choose from several cost-effective development packages. There is even a full range of supporting products, including a symbolic debugger, resource editor, utilities and specialized libraries.

You'll discover that your software purchase is backed by an excellent warranty and skilled technical support staff. You'll appreciate having access to LBBS—one of the world's first 9600 baud, 24-hour bulletin board services. And you'll be able to conference with other Lattice users through the Byte Information Exchange (BIX) network.

**Cross the finish line.** Order your copy of the Lattice AmigaDOS C Compiler today. We'll supply the speed. You bring the running shoes.

Lattice

Subsidiary of SAS Institute Inc.

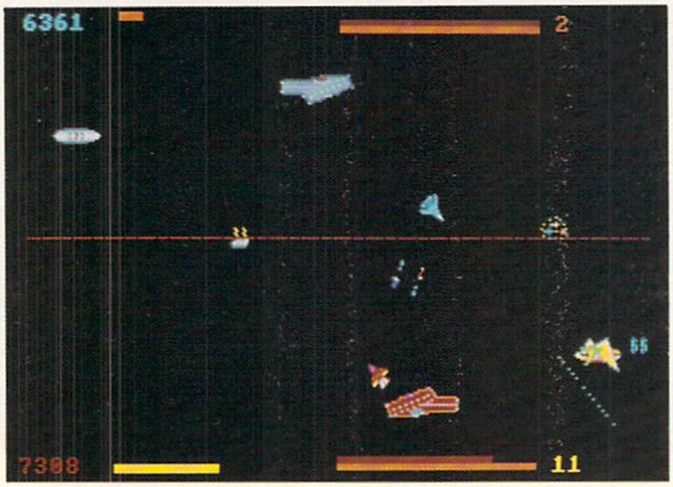
Lattice, Incorporated
2500 S. Highland Avenue
Lombard, IL 60148
Phone: 800/533-3577
In Illinois: 312/916-1600

Lattice is a registered trademark of Lattice Incorporated. Amiga is a trademark of Commodore-Amiga, Inc.
Manx is a registered trademark of Manx Software Systems, Inc.

# Take Five!

**By Steve Hull**
**GEnie: LIGHTRAIDER**
**People Link: St.Ephen**

It's ironic that, while a computer may be dismissed as "just a game machine," few applications require more sheer computing power than do the graphics and sound inherent to a good arcade game. Any MS-DOS machine can handle a spreadsheet, but playing a game on one is like a trip back to the days of Mattel's Intellivision (if the computer even offers color). The Macintosh may be a desktop publishing wiz, but its animation capabilities are embarrassingly primitive. Of course, neither of these computing giants offers much sound besides "plinks" and "beeps," without expensive third-party add-on hardware.



**Libyans in Space**

The Amiga is unique. It can handle serious applications, such as spreadsheets and desktop publishing, simultaneously. When the evening's work is done, the Amiga turns into a game machine superior to all but the most advanced coin-ops, with all the color, action, and sound that game gurus expect. It took a while for programmers to make the Amiga shine, but the fruits of their work have started to flood the marketplace. Home gaming will never be the same.

Of course, that ability makes it tougher to decide which games are worth taking home. That's what this article is all about. On to the games!

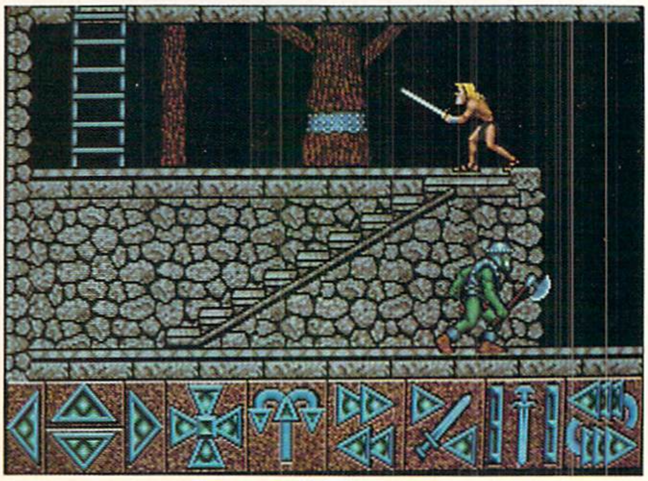
"Welcome to our world," croaks a voice so sonorous and evil that it makes Darth Vader sound like Little Bo Peep. So begins *Libyans In Space,* the debut offering of T & L Products.

*Libyans* just may be the first certifiable New-Wave game—let me explain.

According to the prologue in the documentation, amorphic superior beings orbiting the earth in a strange space ship observe our own U.S. Navy mixing it up with Mohammar's marauders (yes, it's the old Amorphic Superior Beings scenario ... [sigh]). SAMs trace icy fingers toward fighter aircraft, small attack boats get blown to toothpicks by destroyers—the original lousy time had by all. "This is not a fair game!" exclaims one of the Beings. With a wave of his hand, he creates a more balanced simulation in which the two sides may more equally settle their disputes. *Libyans In Space* is that simulation.

*Libyans In Space* is, quite simply, the wackiest arcade game I've ever seen. The game is not outstanding. It's a basic combat simulation—each side is equipped with aircraft carriers, helicopters, fighter jets, frigates, and cruisers. You can represent either the Libyan side or the U.S. Navy, with one-, two-, or no-player modes (the computer plays both sides). The object of the game is to berserk the kapok out of the other guy (just like real life).

This dry description does nothing to portray what an experience playing *Libyans* really is. The action—



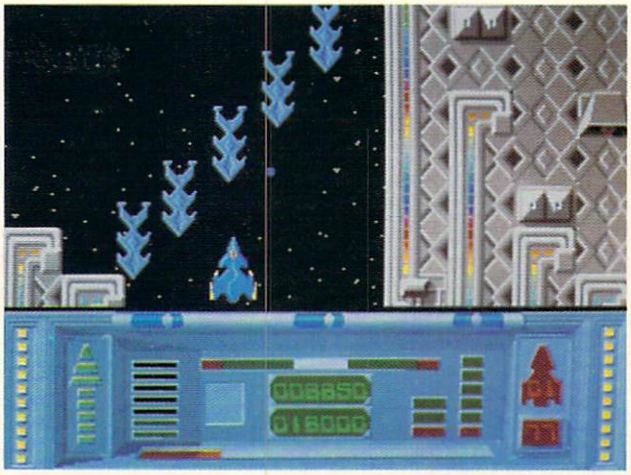
**Barbarian**

especially against the computer—is fast and frantic. Each side unleashes barrages of particle beams, guided missiles, and other congenial stuff.

The graphics are, for the most part, sparse and functional, harkening back to earlier offerings for the Apple II. The sound is another story. Mixed amid the "whup-whup-whup" of choppers and the whoosh of jets are car horns, dog barks, and even what sounds like one of the game's developers yelling in the bathroom.

While all this is going on, an urgent, hyped-up woman's voice keeps up a steady play-by-play: "LIBYA CRITICAL! U.S. NAVY CRITICAL!" She often takes off on an impromptu rap a la Max Headroom: "L-L-L-LIBYA CRITICAL!"

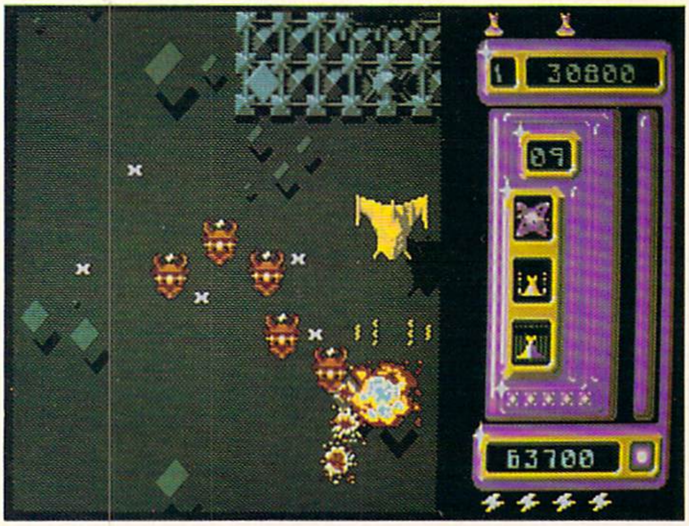
Many play options, including autopilot, fire-and-forget missiles, and the ability to change craft in mid-battle, make *Libyans In Space* an engaging challenge. The biggest challenge I faced was trying to shoot straight as I fell off my chair laughing at the game.



**Insanity Fight**

**LIBYANS IN SPACE**
*T & L Products*
*2645 Wilson Street*
*Carlsbad, CA 92008 (619) 729-4020*
*List price $29.95*
*Copy protection: Documentation-based*
*Graphics: Fair*
*Playability: Fair*
*Sound: Totally wacko!*

*Overall: A decent game which makes clever use of an attention-catching theme. What can I say? It worked. 1 or 2 players.*



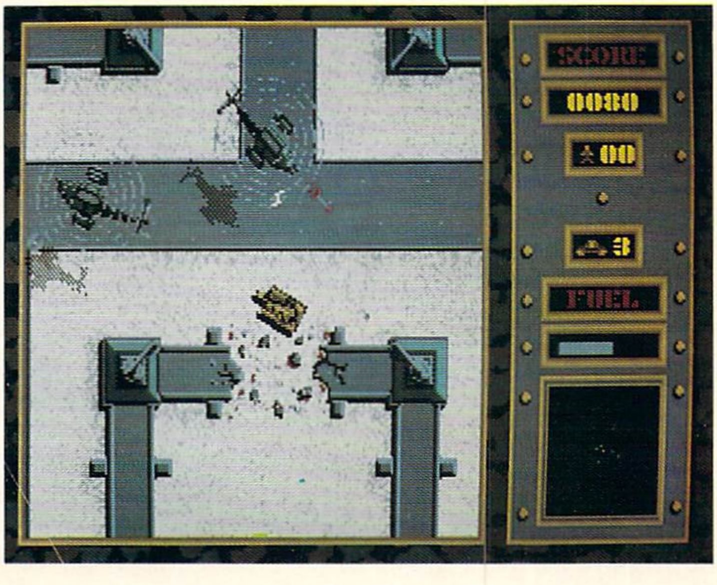
**Goldrunner**

IF IT MOVES, SHOOT IT. IF IT IS STILL, BLAST IT. No, these are not blurbs from the cover of *Libyans In Space II*. Rather, these belligerent blurbs are from the attract-mode instructions to *Goldrunner*, a European import marketed by Microdeal U.S.A. Microdeal U.S.A. is the new banner of Michtron, a company that, until now, has specialized in Atari ST software. In fact, Microdeal U.S.A. included a flyer of Atari ST software with the Amiga version. Optimists, I suppose.

*Goldrunner* is one of those "drop-your-jaw-gorgeous graphics and silky-smooth animation" jobs that snatched more than their share of quarters in the heyday of the arcades. The Amiga translation is seamless; only the cigarette burns on the console near the joystick are missing. Perhaps that description doesn't do the game justice; the graphics are actually better than most coin-ops.

You play the part of the Goldrunner, a Han Solo-type recruited to subdue the treacherous Ring Worlds of Triton, which just happen to

*(continued)*



**Firepower**

be populated by the most hostile race in the universe. You must clear the way for refugee Earthlings. Of course, you can't expect the residents of Triton to cooperate. They won't.

The action consists of your escapades in a shiny gold spacecraft, knocking out ground installations, while fending off air defenses weaving toward you in tightly choreographed formations. Colliding with the crafts isn't a problem, but their pulsing disrupter bombs are. The bombs gravitate towards you, slowly but surely.

You can outrun nearly anything. A click of the left mouse button shifts you to Turbo Boost. The ground scrolls by much too fast to see. That turns out to be no small drawback; the taller structures on Triton (identified by their long shadows) are sturdy enough to pound your craft into gold dust. If you survive long enough to deplete the energy resources of one Ring World, you enter a space-battle sequence on the way to the next World. Top scores are retained for each session, but not saved to disk.

As far as playability goes, Goldrunner's mouse control is a little loose. I needed an awful lot of desk to allow for the movement the game requires. I'm not one of those die-hard purists who insist all arcade games must be played with a joystick—Firebird's *Starglider* is a good example of a game enhanced by superior mouse-control—but *Goldrunner* might be more enjoyable with a joystick option.

Upbeat Kraftwerk-ish music accompanies game play. The music quickly becomes monotonous, but can be toggled off with a function key. If only the same could be said for the digitized voice that continuously comments throughout.

"Welcome to Goldrunner," the voice booms at the outset. That's all right, except the voice *continues* to drop comments regularly—which gets a little inane after a while. "Be careful!" it cautions right after you smash broadside into a building. "Well played!" it congratulates at random intervals. "You're dead," it accuses at the end of a game. Perhaps the oddest comment of all is its sweeping, "WELcome!" intoned right in the middle of an all-lasers-blasting dogfight. The nicest thing I can say about this voice is I wish there was some way to turn it off.

While the game doesn't need a lot of documentation, it needs more than what's been provided. Much information is contained on the game's Status Screen Display, but this display is not reproduced in the documentation. The short paragraph of explanation in the manual is not sufficient.

Amiga 2000 owners should be aware of a problem that exists between *Goldrunner* and some A2000s. Due to the way the game reads the keyboard, its response to keypresses (for instance, to pause or reset a game) is painfully slow—up to a full minute, to register a keypress. That's pretty annoying when the phone's ringing and you want to pause the game. It's more annoying the first time you try to enter your name in the high score screen, and it takes ten minutes for the game to resume! It does not affect all Amiga 2000s, so my advice is to try before you buy. A bug fix is in the works.

*Goldrunner* is a lot of fun and quite playable, but lacks the extra touches and polish needed to ensure long-term interest. The shoot-'em-up theme could be done better, and has been, in another Microdeal title: *Insanity Fight.*

**GOLDRUNNER**
*Microdeal U.S.A.*
*576 South Telegraph*
*Pontiac, MI 48053 (313) 334-8726*
*List price $39.95*
*Playability: Good*
*Graphics: Very good*
*Sound: Undistinguished*
*Copy protection: Disk-based*

*Overall: Lots of visual appeal and action for the arcade fan, but this one tends to get old fast. 1 or 2 players.*

From the moment you boot Microdeal's *Insanity Fight*, you know it's something special. The opening screen is an out-the-viewport shot of space, with multi-plane animation, giving a very real feeling of three-dimensionality to the starfield as it rushes by. Game credits and an introduction scroll over smoothly in blue-chrome letters so sharp they appear to be genlocked against the background. A haunting synthesized note builds to a crescendo, then a crash at the game's title. Whew. Gimme air!

The game is no letdown, either. Little extras abound. At the beginning of each game, a Star Fighter appears, his face staring at you from inside his helmet. He winks, and his visor closes. "Star Fighter ready," he declares, and the game begins.

*Insanity Fight* is, in many ways, similar to *Goldrunner*. The object is to survive from one level to the next, while shooting up the landscape that scrolls smoothly under your craft. You must also avoid the nasties that wheel maliciously about.

The similarities end there. *Insanity Fight's* "look" is brighter and more colorful than *Goldrunner's* often macabre dark tones. You get the impression the designers took extra time building visual appeal into even the smallest details. Also, unlike *Goldrunner*, many of the ground targets in

*(continued on page 12)*

# MicroBotics means Amiga™-Power!

Whichever Amiga you own –or plan to buy– we have the expansion you need

## *For the Amiga 2000...*

### HardFrame/2000

Super Speed DMA SCSI Interface

If your application calls for super-speed uninterrupted access to your harddisk, HardFrame/2000 is your answer. This is a high-end, no holds barred SCSI interface that operates at bus speeds. With cable pinouts designed for compatibility with low cost Macintosh hard drives, one HardFrame/2000 can support up to seven devices. Word-length data transfer, FIFO buffering, true DMA, all mounted on a metal frame suitable for mounting standard SCSI 3.5" drives "hard-card" style (or, if you prefer, cable connected to a bay mounted or external disk). Available March/April. Suggested List price $329.

### SB2000 Adaptor Card

StarBoard2 Portability

Large numbers of MicroBotics StarBoard2 owners have moved over to the A2000. To protect their investment in our technology we've made available a simple, low-cost adaptor card that permits the installation of a "de-cased" StarBoard2 inside the Amiga 2000 (in the first 100-pin slot). When adapted to the 2000, StarBoard2 is still fully functional autoconfiguring memory plus you get access to all the StarBoard2 MultiFunction options– StickyDisk, Math chip, parity or the new SCSI Module. Available now. Suggested list price is only $49.95.

### 8-UP! FastRAM

Maximum Memory in One Slot!

The FastRAM card that every Amiga owner will eventually come to –why limit yourself to the possibility of only two megabytes per slot when 8-UP! will take you all the way to the top of the autoconfiguration memory space of EIGHT MEGABYTES ! 8-UP! uses an exclusive MicroBotics-designed memory module, PopSimm, that frees the user to install his own, conventional DIP-style DRAM in standard SIMM sockets on 8-UP!. If you use 256k PopSimms you can install two megabytes on 8-UP!; if you install 1 meg PopSimms, you can install eight megabytes on one card! In either case you can install the memory chips yourself for maximum flexibility and mininum cost. 8-UP! will also accept conventional SIMM memory. 8-UP! is a power efficient, zero wait state, autoconfiguring design. 8-UP! will be available 2nd quarter of 88. Suggested list prices start at $195.

## *For the Amiga 500...*

### M501 Memory+Clock

Half a Meg at a Great Price!

As we are all coming to realize, a 1-megabyte Amiga (at least) is a necessity not an option. When you add the inboard 512k memory and clock module to your A500, make it a MicroBotics-brand, plug compatible work-alike. It uses the exact same kind of memory and the exact same clock and battery. And note that just like Commodore and unlike some third-party expansions, we use a long-lived rechargeable Ni-cad battery by Varta– which you'll never have to replace! Set the MicroBotics clock using the same software (on your WorkBench disk) as you use for the Commodore clock. What's the difference? You get to keep $41 compared to the Commodore version. M501 has a suggested list price of only $159.

**MicroBotics, Inc.**

Great Products Since the Amiga Was Born!

811 Alpha Drive, Suite 335
Richardson, Texas 75081
(214) 437-5330

*SOLD ONLY THROUGH YOUR AMIGA DEALER*
*Tell your dealer he can quick-order from MicroBotics directly – no minimum quantity –show him this ad!*

### StarBoard2/500

Two Megs and a Choice of Modules

The premier memory expansion for the A1000 is now available on the A500. In a sleek, redesigned case with an independent power supply strong enough to power StarBoard2 and another A1000-style StarBoard2, all the power and flexibility of this great expansion device is available to you. Up to 2 megabytes of autoconfiguring, zero-wait state FastRAM, MultiFunction or SCSI module capability for either math chip/StickyDisk functions or fast SCSI harddisk interfacing. StarBoard2/500 also has a unique LED diagnostic/confidence light to indicate the powered up state of your Amiga and your expansion memory. Another A1000 style StarBoard2 can be connected to the expansion bus pass-UP (it exits through the top of the case) for a total of FOUR megabytes of memory and two modules. Suggested list price $495 and up.

## *For the Amiga 1000...*

### StarBoard2

The Expansion Product of Choice

The superb memory expansion for the Amiga 1000, still going strong! Up to 2 megabytes of zero-wait state, autoconfiguring FastRAM in a sleek, all steel Amiga-colored case plus the capability to accept either one of two daughterboard modules: the original MultiFunction Module or the brand new SCSI Module. StarBoard2 is powered by the bus (up to two StarBoard2's can be supported by the A1000) and passes it on. Available now; suggested list price $495 and up.

### MultiFunction Module

High Tech at Low Cost

This "daughterboard" installs on any StarBoard2 (all three Amiga models). It features a socket and software to support the Motorola 68881 Math Chip as an I/O device (MicroBotics pioneered this approach on the Amiga –now directly supported in the math libraries in the new AmigaDOS1.3). StickyDisk gives you the most "bullet-proof" rebootable ram disk –its hardware write protection turns the whole device into a solid state, superspeed disk. alternately, parity checking of StarBoard2 memory can be enabled when extra parity RAM is installed. Finally, the MultiFunction Module carries an easy to use battery-backed clock to set your system time on start-up. Available now; suggested list price $99.95.

### StarDrive Module

Speedy, Low-cost SCSI Interface

As an alternative to the MultiFunction Module, all models of StarBoard2 can accept this new hard disk interface. StarDrive affords you cost-effective, pseudo-DMA access to Macintosh compatible SCSI drives and other third-party SCSI devices. Fast, easy to install including driver software and disk diagnostics. StarDrive also has a battery backed clock to set your system time on boot-up. Available now. Suggested list price: $129.95

### MouseTime

The Port Saving Clock

The easiest-to-use, most cost effective implementation of a battery-backed mouse port clock for the A1000. MouseTime passes the port through for joysticks or other devices. Complete with WorkBench software. Available now. Suggested list of $39.95.

*"Amiga" is a registered trademark of Commodore-Amiga. "StarBoard2", StarBoard2/500", "HardFrame/2000", "8-UP!", "PopSimm", "StarDrive", and "MouseTime" are trade names of MicroBotics products.*

*Insanity Fight* shoot back, or present lethal surprises--like the interplanetary equivalent of a mosquito zapper.

With the exception of a digitized rock score in the attract mode, there is no music. The game puts its full audio resources into a rich array of effects right from Star Wars. Put this sucker on the hi-fi and you won't be bothered by Klingons for weeks!

A lot happens in the course of a game; you will not master this one overnight. Many strategic elements keep it from getting monotonous. From time to time, glowing rainbow arrows point the way. Sometimes they are helpful; sometimes they aim you into a wall.

Fly over a starburst and you are invisible—usually more of a handicap than a help. Another symbol "mirrors" your position; that is, instantly sends your ship to the opposite side of the screen (and right into one of those mosquito zappers). Another adds cannons to your fighter, and yet another accelerates you to breakneck speed. One symbol replenishes your energy while draining your score; another does the opposite. To make sure you don't get bored, a leviathan Death Ship periodically looms across the screen like a spaceborne aircraft carrier menacing your craft. You can "take out" the Death Ship, but you'd better be good to do it, Bubba.

*Insanity Fight* is colorful, action-packed, and addictive. Arcade games don't get much better than this one.

**INSANITY FIGHT**
*Microdeal U.S.A.*
*576 South Telegraph*
*Pontiac, MI 48053 (313) 334–8726*
*List price $39.95*
*Playability: Very good*
*Graphics: Excellent*
*Sound: Very good*
*Copy protection: Disk-based*

*Overall: A proven shoot-em-up formula enhanced by many extras and surprises. This one's a winner. 1 player.*

Just when you thought it was safe to go out exploring the marsh, Psygnosis introduces *Barbarian,* a game addictive enough to earn a place on the FDA's Controlled Substances list.

In this game, you control Hegar, a blond-and-muscled Conan the Barbarian clone big on the deltoids, but more limited (one suspects) in IQ. Hegar's village has been ravaged by mysterious evil forces under the control of the nefarious Necron. Hegar decides Necron could use a good, healthy throat-wringing—so off Hegar goes, crashing through the underbrush.

At this point, game play begins. Hegar is a small, but well-detailed character; you can even see his white teeth as he clenches them resolutely. As the game begins, Hegar is armed with only his broadsword and a generous helping of brute force and ignorance. Go ahead and laugh; this game will humble you.

Let me give you an example. In one game, I made it past a giant red killer frog (with pulsing eyes and a wonderful digitized THRRRRUMM sound), two werewolves, a booby-trapped falling stone (a camouflaged boulder that leapt off its ledge and tried to turn me into lunch), plus lizard-men, battle-axe wielding thugs, and the occasional rock-throwing gargoyle. I dodged falling spikes, did a running forward-flip to avoid a breakaway bridge, and found out the hard way it doesn't pay to retreat in this game. At the end, the screen informed me:

YOU FAILED IN YOUR QUEST AND YOUR LIFE IS FORFEIT

YOU HAVE NOT SCORED
YOU HAVE COMPLETED 12% OF BARBARIAN

*Twelve percent?! Say what?*

If you get the idea there's a lot to this game, you're right. Dozens of nasties and traps lay between you and your goal, the power-crystal of Necron. Destroying the crystal triggers the eruption of a neighboring volcano. You have exactly the amount of time it took you to reach the crystal to retrace your steps back to the surface. People who have succeeded tell me the "winner" routine is quite nice. For now, I have to take their word for it.

Control is exceptionally flexible. A row of symbols, each corresponding to a different function (e.g., run, jump, stop, fight, and so on), appears along the lower quarter of the screen. You may point-and-click these buttons, use keyboard equivalents, or mix any combination of the two methods. From what I have heard about getting past the traps at the higher levels, you must use both to get the timing on the moves right. The documentation notes that a joystick may also be used, but joystick play is not recommended.

Control is further enhanced by what Psygnosis calls "context sensitive animation." This means a control may have more than one function, depending upon the situation. Hitting the jump button while Hegar is walking executes a broad jump. If he's running when jump is clicked, he does a forward-flip. You must master both these moves to get very far.

The animation and graphics throughout are exceptional—with lots of color, detail, and variety. Sound is limited to Hegar's grunts, yells, and whoops, plus different sound effects for each monster he encounters. All sound is digitized and very imaginative.

Though the subject is serious, the designers' approach is decidedly tongue-in-cheek. With his overdone posturing, Hegar is every macho fool you've ever seen; even at rest, he swings his sword and flexes his muscles. When you run him into a wall (as you will, believe me) he impacts with a explosive "WOOOOF!" and staggers back, wide-eyed. You really should click on the Retreat button at least once.

*Barbarian* isn't perfect. Play action tends to slow down when several objects are moving on screen at once—not that you really have time to dwell on it, but it is noticeable. Occasionally, when beginning a new game, *Barbarian* lapses into a slow-motion, single-step play which rebooting fixes. The quirks aside, if I were to make one change, I would add a "save game" feature.

Barbarian gets points for exceptional play value and originality—so far, exactly what we've come to expect from Psygnosis. Pick this one up; you won't be disappointed.

**BARBARIAN**
*Psygnosis*
*Liverpool L3 3AB*
*United Kingdom 051 647 8118*
*List price $39.95*
*Playability: Very good, addictive*
*Graphics: Very good*
*Sound: *GRUNT*SNORT*AUGHHHH**
*Copy protection: Disk-based*

*Overall: An involving hack-'n'-slash with more than a few Indiana Jones-type traps to keep you on your toes. You'll be playing "just one more game" till way past your bedtime. 1 player.*

Remember the excitement generated when Atari first introduced the 2600 video game system? Until that time, the only home videogames available were Pong and a shooting-gallery or two. Atari introduced the 2600 with its interchangeable ROM packs, and nothing has been the same since.

Here's one for the trivia buffs: What game cartridge was originally shipped with the Atari 2600?

The game was *Combat*, and, despite its blocky graphics and growly sound, more than one joystick got torqued into oblivion over it. Back then, if anyone had gotten a glimpse of Microillusions' debut One-To-One title, *Fire Power*, they would have thought they were looking at arcade heaven.

THE DIRECTOR™

**Professional display and animation language for the Amiga™**

Envision a creative freedom you've only *dreamed* about. Imagine page flipping, color cycling, text generation, even IFF ANIM animations, all combined *at the same time* on the same screen. Now, from the simplest slideshow to the most sophisticated desktop video production, that dream comes true with the Director.

- Use any IFF images, any resolution, any number of colors
- Fades, Dissolves, Blits, Wipes, Stencils
- Page flip full or partial screens
- Preload images, fonts and sounds up to your memory limit
- Flexible script-based structure
- Basic-like vocabulary: For/Next, Gosub/Return, If/Else/Endif
- Arithmetic expressions, random number generator, variables
- Execute AmigaDOS commands from the script
- Text string and file input and output
- Keyboard and mouse interaction
- Digitized soundtrack module
- Supports HAM and overscan
- Supports IFF ANIM playback
- Built-in drawing commands
- No copy protection
- And much more . . .

DEMO DISKS $10 each
Probe Sequence (512k)
RGB (1 meg)

**The Right Answers Group**
**Department C**
**Box 3699**
**Torrance, CA 90510**
**(213) 325-1311**

$69.95
Check or money order payable to: Right Answers
Plus $3 shipping and handling, Calif. residents add 6.5% sales tax

Amiga is a trademark of Commodore-Amiga, Inc.

*Fire Power* is not a conventional tank game. You can play its two-player mode for hours without squaring off directly. While two combatants may choose to pound away at each other, that is largely a diversion. At its heart, *Fire Power* is a high-tech contest of Capture the Flag.

You begin the game parked snugly in your tank's armored garage, one of the few indestructible buildings in the game. As you rumble out the exit, fortified seige doors open. You are alone in the big, bad world.

The world of *Fire Power* is very big and most assuredly bad. The playfield is several times larger than your screen, and scrolls into view as you explore the rugged terrain. Often-mined roads make forays into the green broccoli-like forest an inviting alternative. Don't stay in one place enjoying the scenery too long—if you do, the enemy locks on your position and sends attack helicopters to deliver armor-piercing calling cards. Ya knew the job was dangerous when ya took it, Corporal.

Your first goal is to locate the enemy compound; in a one-player game, that's not usually much of a trick, as the enemy fort sometimes adjoins your own. That strikes me as a little strange, as does the appearance of mines inside the friendly fort; the work of infiltrators, perhaps.

Most of the time, the enemy finds you before you find him. The enemy compound is defended by scores of automatic turrets, high atop the fortress walls. These blasters track your tank at 45-degree increments. You can use this limitation to carefully approach (from a 20-degree angle, perhaps) and blast them into concrete dust without taking any hits.

*(continued)*

NEW FROM ROBOT READERS
A Powerful New Way To Learn To Read

*The Ugly Duckling*

**ROBOT READERS** Are Designed to Teach Children To Read In An Effective, Positive Way Never Before Possible

ALSO AVAILABLE
* CHICKEN LITTLE
* LITTLE RED HEN
* AESOP'S FABLES
* THREE LITTLE PIGS

SUGGESTED RETAIL: **$29.95 each**
for the Amiga 512k

AT YOUR FAVORITE COMPUTER STORE
Hilton Android
PO Box 7437, Huntington Beach, CA 92615
(714) 963-4584

Once you dispatch the guns, you can knock a hole in the wall and clank through in search of the flag. While inside the compound, you can pick up extra points by freeing POWs.

*Fire Power* is unique in that each of its play modes—one player, two player, and remote modem play—presents a different game. In one-player mode, staying alive is a major consideration; five lives and you're out. Both two-player modes let contestants play a single game until their joystick-hands cramp, so play becomes bolder. In local two-player mode, each player can see the other's screen, which means if you see your opponent getting too close to your flag, you can quickly return home to defend your compound. In modem play, you see only your own playfield. The right side of the screen is taken up by a simple terminal which allows players to type messages while the game is in progress. You can't see what your enemy is up to; he might, for instance, be placing a row of mines right outside the front door to your garage. (Who would even think of doing such a thing? Heh, heh, heh.)

*Fire Power's* graphics are nice and "arcadey." Sound is rich with explosions, artillery barrages, and even a little squish when your tank runs over an enemy soldier. (Programmers think of everything nowadays!)

The documentation comes entirely on the disk, and is very good, aside from some embarrassing misspellings. Pay close attention to it the first couple of times because you can't review it without rebooting. A quick reference card would be a welcome addition.

Modem mode is simple and straightforward, requiring almost no knowledge of telecommunication, as long as your modem is Hayes compatible. If your modem is not compatible, the game is not guaranteed to work in this mode. That's something that should be stated on the package. In practice, however, you should not have problems. Most computer games are more fun when played against a human adversary, and modem mode makes this convenient.

Like other tank games before it, control is a little tricky. Your best bet is a good, positive joystick like the Wico Boss, or, for aspiring Johnny Rambos out there, The Terminator from SuperSoft.

With the introduction of its One-To-One software line, Microillusions promised maximum arcade action and playability at a modest price. *Fire Power* delivers everything promised, and then some.

**FIREPOWER**
*Microillusions, Inc.*
*17408 Chatsworth St.*
*Granada Hills, CA 91344*
*(818) 360-3715*
*List price $24.95*
*Playability: Excellent*
*Graphics: Very good*
*Sound: World War II in your Amiga*
*Copy protection: Disk-based*

*Overall: The new standard for tank battles. Superior game play in both 1 and 2 player modes; modem play is icing on the cake.*

That's it for this month. If you would like to see a particular game reviewed (or just hear some hot gossip), please write to me care of this magazine, or send E-mail to one of the addresses at the head of this article. Until next month ... try not to be a Space Invader.

*Special thanks to San Antonio's The Computer Experience for their assistance in the preparation of this article.*

•AC•

# How to keep an eye on your money.

Haicalc is the best valued spreadsheet available for the Amiga.

If you need to keep an eye on your money, you need Haicalc.

For only $59.95 you receive a powerful tool to project budgets, calculate expenses, prepare a financial statement or tackle any large financial or numeric task.

Haicalc fully utilizes the Amiga Intuition user interface. You have the convenience of simple point & click selection to start from Workbench, pull down menu selections for commands, and economical use of your computer's memory — the program with blank spreadsheet uses only 165k.

You harness the power of multi-tasking, multi-windowing, multiple fonts and styles with a maximum spreadsheet size of 9,000 by 9,000.

Compare this to the competition. Haicalc is a powerful solution for a painless price.

HAITEX

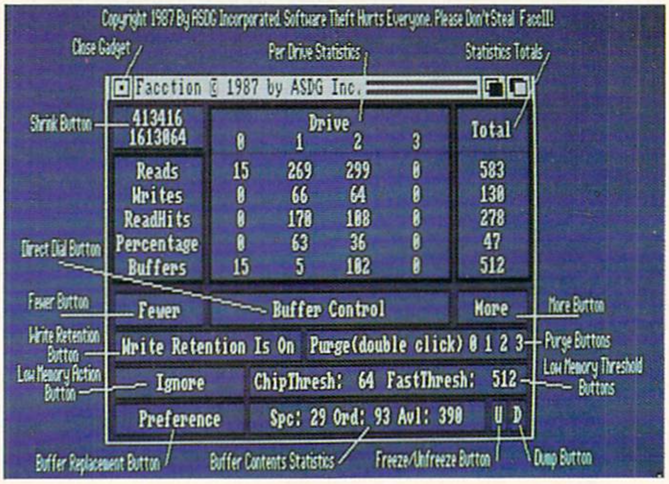
Haicalc is a trademark of Haitex Resources
Amiga is a registered trademark of Commodore-Amiga, Inc.

**HAITEX RESOURCES** 208 Carrollton Park Suite 1207
Carrollton, Texas 75006 (214) 241-8030

# FACC II

by Graham Kinsey
Plink ID: G KINSEY

## Forced to get by with floppy drives? Speed up data accessing with this floppy caching program.



```
Copyright 1987 By ASDG Incorporated. Software Theft Hurts Everyone. Please Don't Steal FaccII!

Facction © 1987 by ASDG Inc.

  413416            Drive                Total
 1613064     0     1     2     3

   Reads    15   269   299     0          583
  Writes     0    66    64     0          130
ReadHits     0   170   108     0          278
Percentage   0    63    36     0           47
 Buffers    15     5   102     0          512

  Fewer         Buffer Control             More
Write Retention Is On  Purge(double click) 0 1 2 3
  Ignore     ChipThresh:  64  FastThresh:  512
Preference   Spc: 29 Ord: 93 Avl: 390       U D
```

What do you do if you don't own a hard drive? Do you have to suffer with long access delays each time your floppy has to fetch yet another piece of data? Are you tempted to take a nap every time you type "dir opt a"? Does your database program appear to be recovering from a hangover every time it has to display records? There is a solution for Amiga owners who have to get by with floppy drives, but don't want to significantly reduce the amount of time they can actually use their Amigas. The answer is FACC II (Floppy ACCelerator) from ASDG Inc.

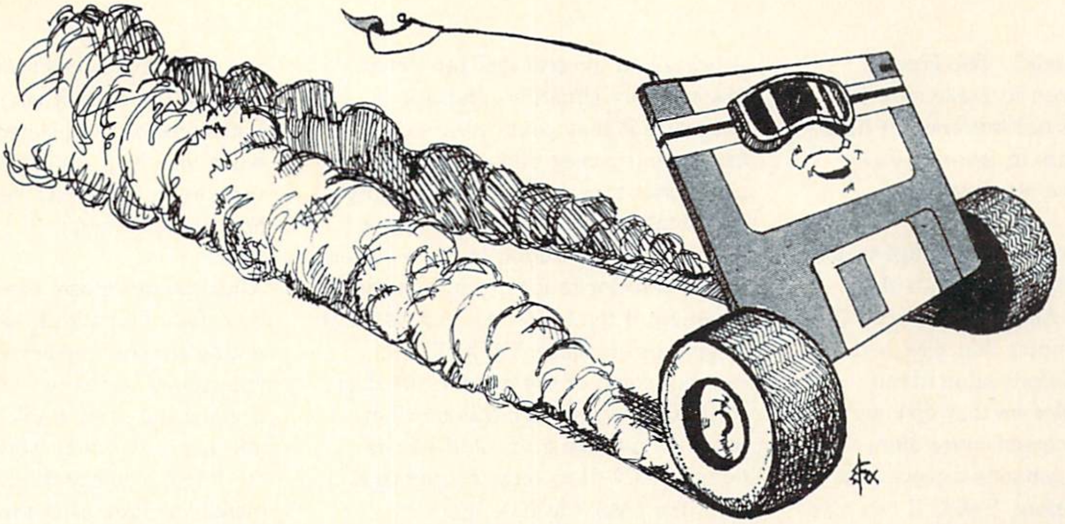
FACC II is a floppy caching program. Disk caching uses memory to improve the speed at which data is accessed from a disk drive. When a sector of data is first fetched from a disk drive by some program, the data is stored in a portion of memory provided by the caching program. So the next time that sector of disk data is needed by the program, the disk drive does not have to be accessed. Instead, the sector of data is fetched directly from the cache. As a result, disk caching saves a lot of time and wear-and-tear on your drives.

Before we discuss the features of FACC II, let's review FACC, its predecessor. FACC is also a floppy caching program, but it offers little more than the cache itself.

FACC's user interface consists solely of a small window, in which information about how the cache is operating is shown. The only control the user has over FACC is the amount of memory, in terms of buffers (one buffer being 512 bytes, the same as an AmigaDOS floppy disk sector) one could allocate to FACC. Even worse, FACC cannot be controlled from the CLI whatsoever. You cannot even load FACC from a startup-sequence and tell FACC how many buffers it can have via the startup-sequence. Although FACC does have a shrink button to reduce the size of the FACC window, you must have that window present to run FACC. FACC also does not know the structure of an AmigaDOS disk, making FACC less efficient than it should be. FACC treats every disk sector with equal respect; when you reduce the allocated amount of buffers (forcing FACC to give up buffers), FACC just throws out the oldest one.

FACC is very incompetent with formatting and copying disks. You cannot be sure that the disk you just formatted with FACC is clean. The program is more destructive when performing a diskcopy; FACC often fills the destination disk with trash. FACC also does not know the true name of each floppy drive (df0:, df1:, df2:, and df3:). The first time you insert a disk into a floppy drive after loading FACC, that drive is automatically labeled drive "A" (the drives could be labeled A, B, C, or D), even if you insert a diskette into one of the external drives first (i.e., not df0:). This makes the statistics information in FACC seem confusing, since you might assume that drive A is df0:. Of course, the tremendous savings in time produced by FACC via its disk cache more than offset any problems created by it. It is still a wonderful program, especially for a first version. (ASDG certainly didn't release a product to the Amiga marketplace that still needed beta-testing.)

FACC II is a great improvement over FACC. Many problems, large and small, have been addressed. One major change has to do with FACC II's user interface. The caching program itself has no user interface. When you first execute FACC II, it buries itself in the operating system,



never to be seen again. Instead of FACC's user interface window, FACC II has a totally separate program that serves a dual purpose. In addition to providing the user with a display and information window, it also serves as the user interface, allowing the user to communicate with FACC II.

Actually, FACC II comes with two such programs. The first version, called SatisFacction, duplicates FACC's user interface. This allows old users of FACC to start using FACC II immediately, without having to learn most of the new features. The second version is called Facction. Facction's window is much larger than FACC's and SatisFacction's display windows because Facction allows the user to access the new features found in FACC II. The top half of Facction's window appears to be identical to FACC's and SatisFacction's windows.

There have been a few important changes between FACC's user interface and SatisFacction (and the top portion of the Facction window). The shrink/ expand button now displays the amount of memory available when the Facction window is in shrunken form. Free memory is now separated into CHIP and FAST memory, instead of being lumped together. The drive labels are now 0, 1, 2, and 3, and these labels now directly correspond with floppy drives df0:, df1, df2:, and df3:.

The Fewer and More buttons, which were the original way to control the number of buffers allocated to FACC/ FACC II, now operate more efficiently. If you hold either button down long enough, the increment count jumps from one buffer to ten buffers at a time. The buffer control bar, which used to do absolutely nothing, now serves a purpose. If you click on the buffer control bar, the words "Buffer Control" are replaced by an integer gadget, with the amount of currently allocated buffers inside the gadget. You can alter this number, and immediately change the number of buffers allocated. (This is referred to in the FACC II manual as "direct dialing.") All the changes in the top half of the Facction display also apply to SatisFacction's display.

The bottom portion of the Facction display (the portion that SatisFacction lacks) provides access to many of FACC II's new features. One new control box is the Write Retention box. When a sector of data is written to a floppy disk, FACC always rereads that sector of data into the cache, even if the block of data had never been in the cache before. In FACC II, you can choose whether new sectors of data should be retained in the cache, or if the sector should only be kept if it was in the cache to begin with.

Whether Write Retention would ever matter to you depends on your situation. For example, if you have just saved a record from a database to disk, you obviously want the sector(s) of data retained in the cache, so you quickly access that record again. But if you are making a backup of a text file via the AmigaDOS copy command, you may only want to access the original file once it is backed up, without touching the backup.

Next to the Write Retention box is the Purge box. Whenever you eject a floppy disk, all buffers belonging to that floppy are immediately ejected from the cache (more on the consequences of this later). However, you may want to flush a floppy's buffers while the disk is still in the drive. For example, if you just loaded a huge program that you know you will not be loading into memory again (especially if you plan to keep the program running while you work on other tasks), why waste all your buffers on that program? The Purge box allows you to automatically flush the buffers that belong to one floppy drive.

There are also two small buttons in the Facction display that are related to purging. The Dump button tells FACC II to give all allocated buffers that don't currently hold any disk sector information back to the free memory pool. The Freeze button, when activated, tells FACC II not to allocate any buffers to disk sectors that

*(continued)*

have just been read. The Freeze button allows you to make sure FACC II never throws out buffers that hold older disk sectors in favor of disk sectors that have just been read.

FACC II is much more intelligent than FACC. FACC II understands the structure of an AmigaDOS floppy disk. FACC II also knows that disk sectors which contain information about directories or files on that disk are bound to be accessed more often than a sector which contains a piece of a data file. Therefore, FACC II can give special treatment to these sectors. The Buffer Replacement Button in Facction let's you tell FACC II to give special treatment to buffers which contain disk sectors with file or directory information. With the Buffer Replacement Button set to "Preference," FACC II throws away buffers which hold "ordinary" disk sectors first.

Facction has a window next to the Buffer Replacement which tells you how many of the buffers allocated to FACC II hold special disk sectors, how many hold ordinary disk sectors, and how many buffers are empty (available to hold disk sectors).

FACC II also has better manners than FACC. To be more precise, FACC II is not selfish with memory—the most highly valued resource in the Amiga environment. Without memory, no program can meet its potential. The multi-tasking capabilities of the Amiga often cause free memory to be gobbled up by programs more quickly than seats during a game of musical chairs! To make matters worse, FACC II is the type of program that can never have enough memory. The drive access speed increase that you receive from using FACC II is directly related to how much memory you allocate to FACC II for its cache.

So what can be done to relieve an almost constant lack-of-memory crisis in an Amiga that is running FACC II with other programs? ASDG's Low Memory Server (LMS) is a program (it is actually a library) that can detect low-memory situations and ask programs if they could please deallocate memory they don't really need, so a program that really needs memory can obtain some. Clicking the Low Memory Action button in Facction tells FACC II to lookout for a low-memory situation; if the LMS does ask FACC II to give up memory, FACC II will sacrifice some of the memory used to store disk sectors so a less fortunate program can use it. In addition to telling FACC II to monitor messages from the LMS, the Low Memory Action button can determine whether FACC II should give up buffers indiscriminately, or give up buffers holding ordinary disk sectors first.

There is another portion of the Facction window that deals with low memory situations. This is the area devoted to the Low Memory Threshold buttons. The LMS is actually nothing more than an alarm, which is set off whenever any task fails to make an AllocMem() call. Since the LMS is not intelligent, you can tell FACC II what it should consider to really be a low-memory situation, and, therefore, when it should accommodate the LMS and when it should consider the cries of the LMS to be a false alarm. The Low Memory Threshold buttons are actually two integer gadgets that allow you to define, in terms of both CHIP and FAST memory, the critical amounts below which FACC II should give up memory when the LMS asks it to. Feel free to allocate as much memory as you have, because with the Low Memory Action feature, you can be sure that FACC II will not disable a program that needs memory.

There is one feature that I haven't covered yet: the CLI interface portion of Facction. Every feature that can be accessed via Facction's display window can also be accessed via the CLI. Besides the obvious convenience to those who like to use a command-line interpreter (or hate to use the mouse), this now makes it possible to control FACC II from a batch file. You can have all the default settings of FACC II changed via the startup-sequence when you boot up. (Only Facction supports a CLI interface; SatisFacction has no CLI interface.)

After extensive use, I can find only two small faults with FACC II. When setting the low memory thresholds, you cannot set either value below 16K. If you have some FAST memory, but not much, you may not mind if all of the FAST memory disappears completely, as long as there is a decent amount of CHIP memory left. Therefore, you are forced to set your CHIP threshold 16K lower than you would like to. I don't think this problem will cause any major headaches, because when you have at least a megabyte of memory, 16K is little more than a drop in the bucket. (Those with 512K can ignore this problem completely, since they don't have any FAST memory to begin with.)

The second problem is that if you use the direct dial feature for changing the number of buffers allotted to the cache rapidly (several times in succession), FACC II produces a Guru. Fortunately, this doesn't happen often.

I can't find anything else wrong with FACC II. It does its job extremely well! I do have a long list of features that I would like to see in the upgrade to FACC II. One problem concerns only those who use the Amiga for more than a few minutes before they shut it off, and those who have only one floppy drive: Every time you eject a disk from a drive, all the buffers associated with that floppy are lost forever. The Freeze button does NOT prevent this. So the more you swap disks, the less effective FACC II becomes, since the buffers that are built up are constantly thrown away.

It is apparent that FACC III should be able to remember the volume name of a floppy disk. This would allow the user to request that a floppy's buffers be put in storage while the user changes disks. Equally nice would be the option to freeze all the buffers in the cache, so that even when a disk is ejected, its buffers will still be retained in the cache (assuming, of course, that FACC III knew the volume name of the disk). Once this has been implemented, the next logical step is to create a special device for the cache (similar to a recoverable RAM disk), so that when you reboot the machine or encounter the Guru, your buffers will not be lost.

Another nice feature would allow FACC III to automatically increase its buffers in some way, if there is plenty of free memory. Since FACC II can automatically give up memory if it's scarce, why not give FACC III the ability to allocate more memory to the cache if free memory is plentiful?

A great feature for FACC III would be the ability to fetch certain sectors of a floppy disk, upon your request, before a program that needs data actually forces the floppy drive to read that particular sector. Perry Kivolowitz, president of ASDG, has stated that one of the planned improvements in FACC III will be the ability to perform look-ahead, which means that FACC III would actually load sectors that it thinks will be accessed soon into the cache before they are needed by a program. But it would be even better if the user could *tell* FACC III to fetch a sector of data, since the user knows what he will be doing next, and FACC II can only guess.

Before I conclude, let's examine a small debate. When I mention FACC II to other Amiga users, they quickly say, "I presently use a RAM disk on my Amiga. Why should I use FACC II?" Well, let's take a look at the controversy of using FACC II versus using a RAM disk. First, both solutions use system memory to speed up floppy disk access. (Of course, you can use a RAM disk for a hard drive, but since FACC II doesn't apply to hard disks, let's leave that for now.) That is where the similarity ends. The biggest difference between a RAM disk and FACC II is that a RAM disk deals with whole programs on a disk, while FACC II deals with the individual sectors of a disk.

What does this mean? This means that a RAM disk is an all-or-nothing affair. Either you have the program in the RAM disk, or you don't. On the other hand, FACC II is more flexible. If you have 150K available in your RAM disk, and you want to put a 200K program in there, you simply can't do it. But if FACC II has only 150K in buffers allotted to it, it can keep most of the program in the cache. The next time you want to run the program, very little disk access will be required, since a majority of the program's data is already in the cache.

Of course, when you are able to fit a program in the RAM disk, you always know the program is there. You don't have that same level of confidence with FACC II, unless you use the Freeze button to make sure the buffers that are related to a certain program don't get kicked out of the cache. Another distinction that should be made is, while both methods require that the program be loaded from disk once before any benefit can be gained, FACC II allows you to load the program as an executable file, not just as a disk file in memory. I realize that this has little benefit for those who just load their CLI commands into memory, but if you also load in regular programs that you will be executing in the near future, FACC II will allow you to run the program as soon as you load the data the first time.

For those who like the mouse, FACC II always can be controlled using the mouse with Facction and SatisFacction. A RAM disk is much harder to control via the mouse, because not all programs and commands have icons associated with them (and if you don't have Workbench loaded, you can't control the RAM disk at all with the mouse).

FACC II has many features that allow the cache to be manipulated, while a RAM disk offers users little or no control. The default RAM disk on the Workbench disk has no control features, while ASDG's Recoverable Ram Disk (the most popular public domain RAM disk program) has a couple of commands to allow manipulation of that RAM disk. RAM disks still have features not found in FACC II, like the ability to survive a warm boot. In addition, a RAM disk program takes up much less room than FACC II. A RAM disk can also be easily loaded up with programs and commands via the startup-sequence.

I won't make any recommendations on which you should use, since your decision will have a lot to do with the amount of memory you have, the type of programs and commands you use, and how you use your Amiga in general. I hope everyone realizes that using a RAM disk is no reason to write off FACC II.

FACC II is worth much more than the $35 it retails for. It works very nicely, and is very flexible and easy to use. FACC II is designed for everyone, from the new Amiga user to the experienced programmer. For new users, FACC II contains lots of documentation and icons for the Workbench. If you are a programmer, and you don't like the feel of either Facction or SatisFacction, FACC II comes with plenty of programmer documentation explaining how you can write your own user-interface for FACC II. How many commercial programs have you seen that allow users to rewrite part of the program to suit their tastes? I hope you'll take a look at FACC II, because it may do you a world of good.

•AC•

Fox

# UNINVITED

by Kenneth E. Schaefer

The flaming wreck of your car lies a few hundred feet down an embankment, brightly testifying to the fact that you are stranded. To make matters worse, it looks like a storm is coming up, and the only shelter within sight is a rather gloomy-looking Gothic mansion just up the path. To top things off, your little brother is missing. He wasn't in the car with you when you awakened after the wreck (you checked), and he's nowhere in sight. Maybe he's inside that spooky house. Although you really don't want to, you walk up and knock on the door.

So begins Mindscape's newest graphics adventure, Uninvited. The game is based on the system that made Deja-Vu such a great success. ICOM Simulations, Inc. has developed another winner with this chilling tale of haunted house exploration. Digital sounds (The lightning made me look outside the first time I played!), excellent use of Intuition and Workbench features, and an enthralling plot make this a very playable game that will challenge and delight you for hours.

If you haven't played Deja-Vu, a word of explanation is in order so you have an accurate feel for Uninvited's gaming environment. Your screen is divided into several windows. Each represents a part of your environment, and you can interact with all of them via the mouse. Of course, a large window displays what you see. The graphics are very nice, with lots of subtle detail. There is a window with gadgets that perform the functions you'll need while searching for your lost brother. Another window shows all the exits from your present position. Clicking on any of the exits causes you to leave where you are and go to the new spot. The inventory window displays everything you've picked up during your explorations. When you come upon something you want to pick up, just click on it and drag it into the inventory window. What could be more intuitive?

**"With a dark, gloomy evening to set the mood, you should be in fine shape to enjoy all that Uninvited has to offer."**

One of the first things that really strikes you about Uninvited is how little "fluff" there is. Something is fluff if it appears in a description of a room, but you cannot interact with it. Earlier text-based adventures were filled with fluff. Even some of the more modern graphic adventures include a maddening amount of the stuff. Not so in Uninvited! You can pick up and examine virtually everything in the house. Rugs, paintings on the walls, little boxes in the closets, everything! Granted, you can't take many of these things with you (what a surprise, that chest of drawers won't fit in my inventory!), but if you click on them, and then click on the Examine gadget, you'll get at least a couple of lines of description. It pays to click on everything, examine it, and even open it. You'll be surprised by how many useful things are inside other things.

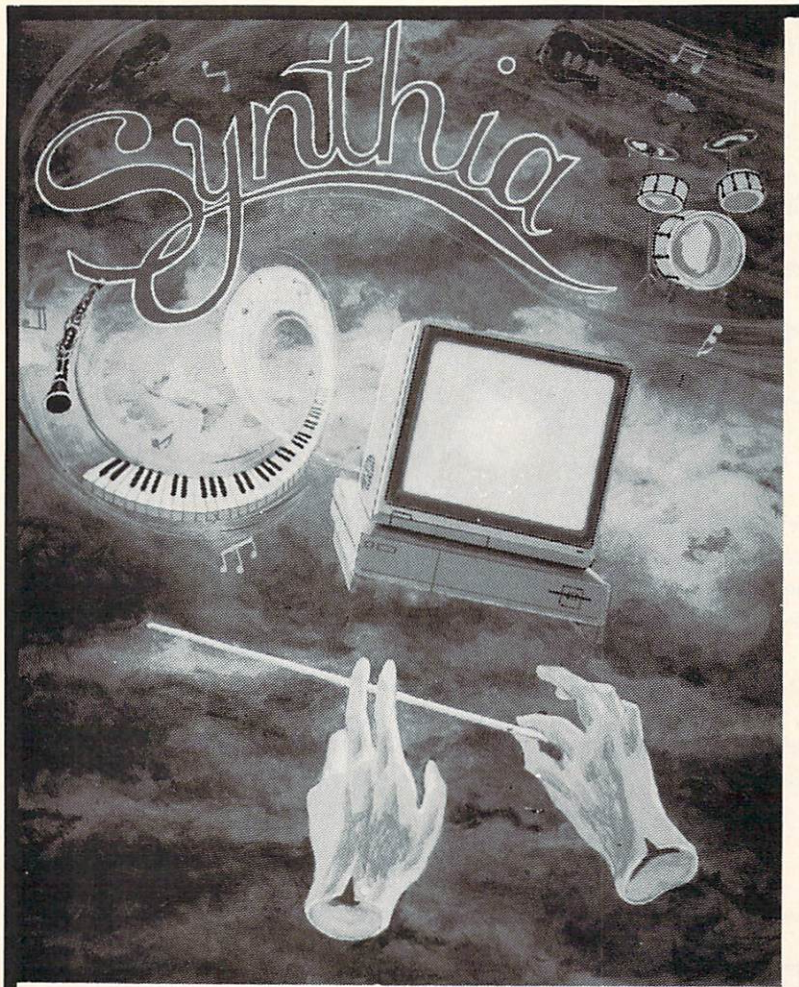
Having said all these glowing things about Uninvited, I guess it's about time to mention the couple of things that I don't like. *[Stepping onto soapbox.]* First and foremost is its copy protection. I understand the need for copy protection, especially in games, but disk-based copy protection stops no determined pirate, and only keeps legitimate users from making back-up copies. It prohibits putting the game in the Games directory of a hard drive. With the advent of text-based copy protection schemes (such as The Pawn) or dongles (which are a lesser of two evils when compared to disk-based copy protection schemes), it's a pity some companies still muck with the disk. Perhaps if we all do our part to hinder pirating, we'll see the death of this form of copy protection. *[Stepping off soapbox.]*

Another thing that may detract from your gaming enjoyment is one of the very things that makes the game so "Amiga-ized." You can actually only do a limited number of things. Your only possible actions are depicted by the gadgets in the upper right-hand corner initially. (You can move them anywhere on the screen.) Your choices are limited to Examine, Open, Close, Speak, Operate, Go, Hit, and Consume. That's it. You cannot type in commands of your own. In fact, the only typing you'll do is giving your saved file a name and talking to someone. You may not see this drawback as all that bad, but if you are accustomed to a more "rich" command set, you may find it stifling. It's a matter of taste. There is one good thing about a limited command set that's always visible: You never need to worry about forgetting a command or not knowing just the right word.

Back on the positive side, however, I have to say that Uninvited's text is as good as some of the best text-only adventure games. It really didn't strike me until I was well into the game. A good story line and excellent prose have always been the stock in trade of text adventures, because that's all they have to offer. In the past, many graphic text adventures relied on the novelty of pictures to support what was otherwise a weakly-written story. Uninvited does not suffer from this weakness at all. It is surprising that it took me so long to notice how good the text is. Perhaps it was because a graphics adventure game pulls your attention away from what the game is saying. You're too busy examining the picture for hints and clues. This attention to detail is important in Uninvited, since so many things that would have been fluff in another game are important here. I think that is one of the game's strongest points; it really demands your total concentration. This is not a game you will breeze through in a night or two.

Another positive aspect of Uninvited is the Exits window. (Actually, it's a part of the gaming system, because it was true of Deja-Vu as well.) You are no longer forced to guess where you can go and where you can't. The little squares in the Exits window tell you exactly where you are allowed to go.

*(continued on page 24)*



*Other Products from The Other Guys*

**REASON** - a professional proofreading system used by universities and writers around the world to analyze and improve writing. **(Has helped raise students grades when used faithfully.)** **$395.00**

| | |
|---|---|
| **OMEGA FILE** - a *REAL* data base & mail merge | **$79.99** |
| **PROMISE** - the *BEST* high speed spell checker. (Even better than Zing!®Spell) | **$49.99** |
| **KEEP-Trak GL** - general ledger for home or business | **$49.99** |
| **AMT** - amortization program | **$39.99** |
| **MATCH-IT** -teaches shapes & colors (preschool) | **$39.99** |
| **MATH-A-MAGICIAN** - add, subtract, multiply & divide | **$39.99** |

***Call or write for more information.***

# SYNTHIA *High Performance Digital Synthesizer*

A state of the art music tool which will:

Create digital **IFF Instruments** for use with nearly all music programs!

Modifying existing **IFF Instruments.** Use *SYNTHIA* on digitized samples to add reverb, wow, and other enhancements.

*SOMETHING FOR EVERYONE:*

***Additive Synthesis*** - a traditional method which can create almost any type of instrument.

***Plucked String Synthesis*** - simulates plucked strings . . . right down to the 'pluck'.

***Interpolative Synthesis*** - a method which introduces the natural imperfections found in instruments. (Instruments such as brass, woodwinds, pianos, etc.)

***Percussion*** - build your own drum set . . . create any drum you desire.

***Subtractive Synthesis*** - a simple method of creating instruments.

***Special Effects*** - includes filtering, amplification, phasing, waveshaping, amplitude modulation, real reverb, and . . .

***IFF Music Player*** - powerful and compact. Now you can enjoy those songs that needed a memory expansion before! Up to 32 tracks and 32 IFF Instruments! Supports chords, ties, etc.

**NOW SHIPPING**

*IS IT LIVE . . . OR IS IT SYNTHIA?*

Synthia uses the latest technology to generate realistic sounding instruments and even the new families of instruments sound real. A real synthesizer on a real computer!

Why buy digitized instruments when you can *SYNTHIAsize* them?

Requires AMIGA 512K

**$99.99**

Copyright©1987, THE OTHER GUYS Software • AMIGA is a registered trademark of Commodore Amiga

THE OTHER GUYS

55 North Main Street
Suite 301-D
PO Box H
Logan Utah 84321

(801) 753-7620
**(800) 942-9402**

VISA MasterCard

Double-clicking on these little squares results in the action—if the door is closed, it opens; if the door is already open, then you go through it. Such discrimination on the program's part helps remove the play-mechanics as a layer between player and game. It helps make the experience just that little bit more enjoyable.

Little scenes of animation in Uninvited constantly spring up to startle and delight you. One of my favorites is a little black spider that crawls across your view as you stand on the veranda. You cannot interact with these; your input is blocked while the animation is taking place. In that sense, they are fluff, but they add a nice element of realism, which I found quite pleasing and chilling, to the normally static pictures.

I also found another interesting thing to look for while exploring the mansion. Macintosh jokes! This game was originally designed and released on the Mac, and the authors obviously had Apple owners in mind when they hid these little jokes. As an example, examine the phonograph in the Rec Room. A MacNifty label? What's that doing here in an Amiga game? There are lots more like that, and it can be kind of humorous discovering them from time to time.

It's very tempting when reviewing adventure games to give away all sorts of hints and clues. I'm going to resist that urge and instead give these three pieces of very helpful advice.

1. The Operate command may not work exactly as you would expect. To squirt someone with a seltzer bottle (a purely hypothetical situation in Uninvited, but useful as an example), you click on the seltzer bottle (which should be in your possession), click the operate gadget, then click the recipient of the joke. You'll find this sequence *very* useful against bothersome belles.

2. Don't leave a room without Examining and trying to Open everything in sight (even if it seems ridiculous or impossible). Leave no stone unturned. If you miss something, you may find yourself running back with an ugly, nasty creature chomping at your heels. Those times can be very nerve-wracking, and give you no chance for a wrong turn. Do yourself a favor and look carefully at *everything*.

3. Last, but not least, *save! save! save!* Everything happens pretty fast, and it's easy to die, so save regularly. Nothing is more frustrating thanhaving to start over after playing for two hours. Do yourself a favor and save often.

With this good advice in mind, and a dark, gloomy evening to set the mood, you should be in fine shape to enjoy all that Uninvited has to offer. And you'd better hurry ... You haven't seen your brother in a *long* time.

•AC•

**A NEW GENERATION OF CHARACTER QUALITY**

**PRO VIDEO PLUS**™

***Designed and developed by JDK Images, who gave you Amiga's first professional character generator software.***

- ***Full Screen Video (672 x 440)***
- ***16 Colors Per Page***
- ***16 Fonts Resident***
- ***Multi-Colored Characters***
- ***Automated Background Creation***
- ***Advanced Editing Features***
- ***90 Effects Transitions and Line Definable Moves***
- ***100-1000 Pages****
- ***Keyboard Interface***
- ***Imports IFF Pictures****

**memory dependent*

$299.95

**PRO VIDEO PLUS**™ **by JDK Images is a pro-quality program, compatible with all Amiga machines** (one mega-byte minimum required). **To order see your Amiga Dealer; or contact PVS Publishing for more information.**

**PVS PUBLISHING • 3800 Botticelli, Suite 40 • Lake Oswego, OR • 503/636-8677**

Pro Video Plus ©1988 JDK Images, all rights reserved Amiga is a trademark of Commodore-Amiga, Inc.

# Go with the Flow!

# ...an Idea Processor for the Amiga

*by Pamela Rothman Co-Sysop, A.M.U.G. BBS 516-234-6046*

### What Is an Idea Processor?

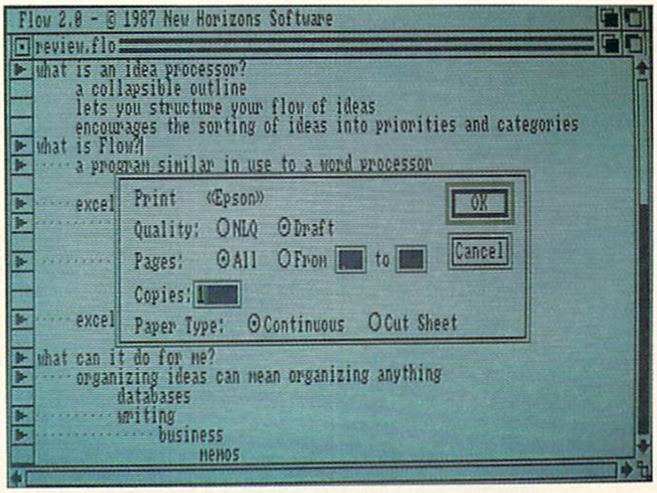
Just as a word processor processes words, an idea processor processes ideas. An idea processor can be useful to anyone, because even people who don't need to process strings of words have ideas they need to organize. An idea processor lets you create an outline that you can expand or collapse to structure your flow of ideas. Remember the outlines you wrote in school? Your teacher was right; outlining helps you sort ideas into priorities and categories.

### Flow's Easy Editing

Flow 1.0 (reviewed by Erv Bobo in AC V1.6) was an excellent program. Version 2.0 is even better.

Flow lets you manipulate headings and subheadings within a seemingly endless hierarchy. It is similar to the way the Amiga stores files on a disk. Flow has the same editing functions as a word processor, and the two interact perfectly to take the ache out of facing a blank screen. Instead of wondering, "What will I say?" you can use Flow to map out a path, putting your thoughts on the screen and arranging them as you go along.

You can use keyboard commands or menus to cut, copy, paste, or delete your text. You don't even have to arrange them in the correct order, since Flow makes it so easy to rearrange them. Flow supports as many as four windows to cut and paste anything from a single word to an entire heading and all its subheadings. Clicking and dragging the mouse highlights text from anywhere on the screen without taking a trip to the menu window first. Text enhancement, such as boldface, italics or underlining, can be added for emphasis. To make it easy to spot related ideas, you can add text enhancement to a heading so that all subheadings generated also carry that enhancement. Version 2.0 of Flow includes some of the best editing aids I have ever seen, such as the option to highlight and delete text by typing substitute words, or by hitting DEL or backspace.



```
what is an idea processor?
    a collapsible outline
    lets you structure your flow of ideas
    encourages the sorting of ideas into priorities and categories
what is Flow?
    a program similar in use to a word processor

    excel
    excel
what can it do for me?
    organizing ideas can mean organizing anything
        databases
        writing
            business
                memos
```

```
Print    «Epson»
Quality:  ○NLQ   ●Draft
Pages:    ●All   ○From    to
Copies: 1
Paper Type:  ●Continuous   ○Cut Sheet
```

### Outlines in Motion

The ability to expand and collapse headings and subheadings is the true heart of this program. (For quick reference, I can flip between Flow and the word processor with left Amiga M and N, since Flow is treated like a Workbench screen.) In the idea tree below, I've listed three to convince my boss to pay for this program.

- get Boss to pay for program
    - explain how it will make me a better worker
    - how I will use it for work
    - so he should pay

As you can see, I have one main heading, "get Boss to pay for program," with three subheadings. The main heading has a triangular symbol in the left margin to show that it has subheadings beneath it. I created the three subheadings by pressing shift/return at the top line, and simple returns thereafter. Each time I hit return, the cursor creates a new line at the same level as the line above it. You can set the program to indent from one to five spaces for each new level.

- get Boss to pay for program
    - explain how it will make me a better worker
    - how I will use it for work
    - so he should pay
    - because I am a marvelous person

Now that I look at it, I see the outline could be changed to present my ideas better.

- get Boss to pay for program
    - explain how it will make me a better worker
    - how I will use it for work
    - because I am a marvelous person
    - so he should pay

The cursor looks like an I-beam, and is referred to as such in the manual. I put the cursor on "so he should pay" and selected Move Up from the Subheadings Menu. Other options are Move Down, Expand, and Collapse. If you collapsed the outline above, it would consist of "get Boss to pay for program," with the other headings hidden beneath.

*(continued)*

**Choosing a New Direction**

I have some fine ideas in my outline, but they need to be better organized. Perhaps this version would work better.

```
get Boss to pay for program
  why he should pay
  because it will make me a better worker
  because I would use it for work
  because I am a marvelous person
```

I highlighted and dragged the heading "why he should pay" to the line just below "get Boss to pay for program." Then I used Tab once on the remaining three lines to move them to a new level. This better expresses my thoughts. I still need to add more subheadings under the three new headings, though.

```
get Boss to pay for program
  why he should pay
    because it will make me a better worker
      I would be more organized at work
    because I would use it for work
      to structure my time and resources
    because I am a marvelous person
      I am always thinking about helping the company
```

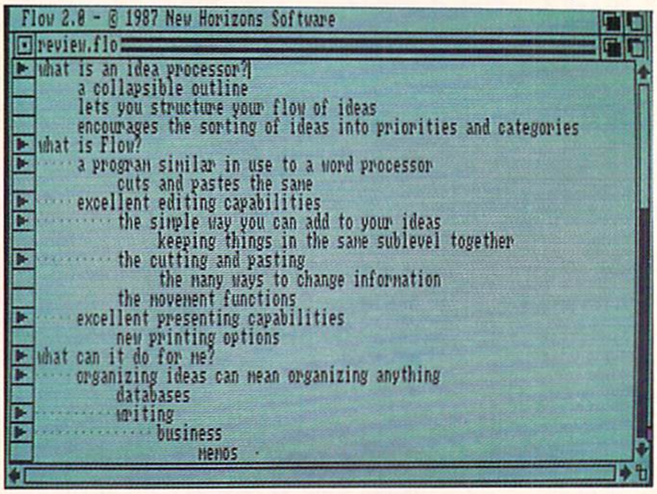
**Eureka!**

So far, I have tabbed over the three headings (ideas) I decided were subordinate to my main purpose. I have also expanded the three reasons why he should pay for Flow. Now I use the mouse to rearrange the headings, suddenly giving a subordinate idea enough importance to change the whole picture.

```
get Boss to pay for program
  why he should pay
    because I am a marvelous person
      I am always thinking about helping the company
    because it will make me a better worker
      I would be more organized at work
    because I would use it for work
      to structure my time and resources
```

If I collapse the heading that reads, "because I am always thinking about helping the company," I get the spine of the memo I will write to my boss:

```
get Boss to pay for program
  why he should pay
    because I am a marvelous person
      I am always thinking about helping the company
```



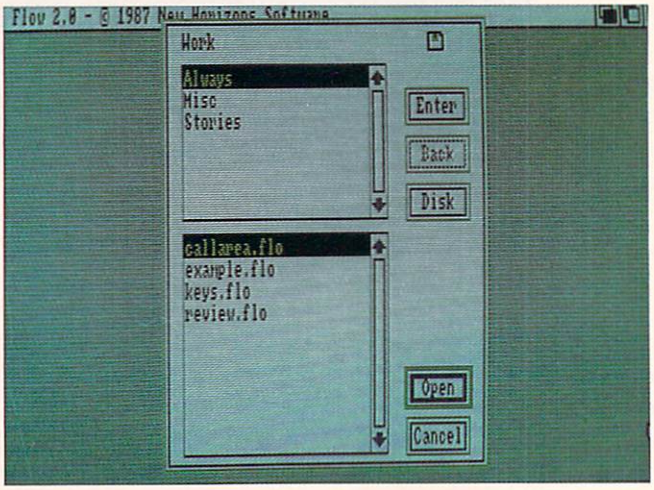
```
what is an idea processor?
    a collapsible outline
    lets you structure your flow of ideas
    encourages the sorting of ideas into priorities and categories
what is Flow?
    a program similar in use to a word processor
        cuts and pastes the same
    excellent editing capabilities
        the simple way you can add to your ideas
            keeping things in the same sublevel together
        the cutting and pasting
            the many ways to change information
        the movement functions
    excellent presenting capabilities
        new printing options
what can it do for me?
    organizing ideas can mean organizing anything
        databases
        writing
            business
                memos
```

You can also use function keys or the menu to create a new outline, open an existing outline, close, print, or even revert to the version saved on disk if you have hopelessly mangled your outline since the last time you saved.

You can sort headings in alphabetical order. The features Find and Find Next work within collapsed headings.

**Features**

Loading and saving is done from the Project menu or by using keyboard shortcuts. The current directory is shown in a window with a gadget to scroll up and down. Clicking the disk gadget automatically moves you to the next logical device on the system, including floppies, RAM, VD0, & HD0.



Press right Amiga/S to save the outline to disk in the same location as the last save. If you haven't named the outline yet, the program automatically brings up the Save window for you. Flow can save the outlines in its own format or in a text-only format. When a Flow text file is loaded into some word processors, the tabs are represented as spaces; this is not usually a problem with Flow.

With Flow's improved printing capabilities, the outlines can stand on their own. You can print with headers and footers, in draft or in NLQ, and the program will skip over perforations. The printing window shows the printer currently selected in Preferences, so you can find out if you have the wrong driver before you start printing. Sending to a printer with the wrong driver can cause all sorts of formatting problems and waste paper. My husband and I have a switch box to let us use either the letter quality or the dot-matrix printer. This welcome feature has saved me twice from printing with the wrong driver.

**Flow's Manual**

The manual is well designed, something I would expect from the company responsible for Flow. (If their own program couldn't get them organized, I would worry about my purchase!) The manual is in a spiral ring binder, so it lies flat for easy reference. The manual is clear and easy to follow. The

*(continued on page 28)*

NEW for the AMIGA

# Precisely™

**The Word Processor that doesn't try to be a desktop publisher.**

Programs designed for graphic artists are difficult to learn, hard to use, tie up lots of memory, cost too much, and do not handle text very well. Graphics printing on a dot matrix or daisy wheel printer is very slow.

Introducing PRECISELY. The word processor optimized for people who work with words not pictures! Everything you expect in a word processor, such as:

- What you see is what you get (WYSIWYG)
- Very fast screen update/printer speed
- Easy to learn and simple to use
- Converts PaperClip, Pocket Writer and SpeedScript documents
- Online HELP • Mail Merge • Keyboard Macros
- Oops Key to undo mistakes
- Built in spelling checker coming soon

This is PRECISELY the word processor you've been looking for at PRECISELY the right price! Only $79.95 (sometimes less is more). Add $3.00 for shipping and handling, CA residents add 6% sales tax.

*Don't fumble around with your Amiga files.*
*Let QUARTERBACK manage your valuable data.*
*The Quarterback sneak scores every time!*

# QUARTERBACK™

*Quarterback is a **FAST** Hard Disk to Floppy Backup Utility for the Commodore Amiga, featuring:*

• Fast backup - 20MB in less than an hour • Full/Subdirectory/Incremental backup • Full/Subdirectory/Single File Restore • Automatic formatting of diskettes • Automatic catalog of files • Automatic diskette sequence numbering and checking • Runs with Workbench or CLI • Includes/excludes files by name and/or wild card • Selects files by date • Selects files by Archive bit • Estimates number of floppies needed • Accepts batch command files • Validates writes • Restores original file date/time stamp • Prints backup/restore report • Allows easy access to subdirectories • Beeps for floppy change • Provides AmigaDOS access from within • Convenient/user friendly error recovery • Multitasking • No copy protection • Works with all AmigaDOS compatible hard disk drives

***You'll have fewer "time-outs" with QUARTERBACK managing your file backups.***

Put Quarterback on your team for only $69.95 plus $3.00 for shipping and handling, CA residents add 6% sales tax.

# TRANSFER FILES

## TRANSFER C64/C128 files to and from your Amiga!

DISK-2-DISK reads your PaperClip, SpeedScript and Pocket Writer documents or other files on floppy disk directly into your Amiga. Transfers all file types. Use these transferred files with your favorite Amiga programs.

- Reads/writes 1541/4040 and 1570/1571 disk formats.
- Converts Commodore/PET ASCII to Amiga ASCII and vice versa.

## TRANSFER MS-DOS and ATARI ST files to and from your Amiga!

**DOS-2-DOS** reads Lotus 123 worksheets, wordprocessing documents or any other files on floppy disk directly into your Amiga for use with your favorite Amiga programs.

- Reads/writes both 5.25" AND 3.5" MS-DOS disks.
- Reads/writes 3.5" Atari ST diskettes (GEM format).
- Converts ASCII file line ending characters.

Disk-2-Disk requires the Amiga model 1020 5.25" disk drive. Dos-2-Dos runs on any standard Amiga. Disk-2-Disk $49.95, Dos-2-Dos $55.00. Add $3.00 for shipping and handling, CA residents add 6% sales tax.

CCS

# Central Coast Software™

268 Bowie Drive, Los Osos, California 93402 • (805) 528-4906 • FAX (805) 541-4745

MasterCard VISA

The Introduction describes how to load from CLI or Workbench, and refers the beginner to the Amiga manual when appropriate. The tutorial, Learning Flow, shows you how to arrange and edit headings. The next section, Using Flow, describes how to sort, cut, paste, etc. There is a complete reference section which includes a list of keyboard shortcuts, an itemized summary of the menus, and an index. I usually found what I was looking for quickly.

I found myself using the manual less often after my first reading because Flow is designed so that a lot of your natural keyboard instincts come into play. A nice touch would have been a function key labeling strip to go above the function keys, but that's just a minor quibble. (I think all software that uses the function keys should provide a simple reference strip.)

### Why Not You?

Organizing ideas can mean organizing anything. Many programs, like databases, require a certain amount of planning. Flow simplifies this process, freeing you from the restrictions of paper and pen. My example of a business idea illustrates how Flow can help you in public speaking, even if it's only an informal discussion with your boss. In only one evening, I wrote a three-page proposal to the president of my company using Flow. I think it would have taken me two or three evenings if I had been using a word processor. I even wrote this review with the help of Flow. With Flow on the computer and a book in front of me, I can easily organize all the pertinent information on the screen, and later sort them by page numbers or subjects. For quick reference, I keep an outline for each piece I write.

What can an idea processor do for you? I predict that Flow will become as indispensable to thinkers as word processing is to writers. Flow could also be the ultimate note-taking program. If nothing else, Flow lets you sit and think in front of the computer, and then have something to show for it. Programs like Flow make the Amiga a boon to the creative person. Flow is the first idea processor I have used, but I don't think my infatuation with the *idea* has blinded me to the *program*. With the release of Flow 2.0, I think New Horizons Software has created a classic.

•AC•

**Flow 2.0**
New Horizons Software
P.O. Box 43167
Austin, TX 78745
(512) 329-6215

$99.95 List

ADD TO THE POWER OF YOUR PROGRAMS WHILE YOU SAVE TIME AND MONEY!

CBTREE does it all! Your best value in a B+tree source!

**Save programming time and effort.**
You can develop exciting file access programs quickly and easily because CBTREE provides a simple but powerful program interface to all B+tree operations. Every aspect of CBTREE is covered thoroughly in the 70 page Users Manual with complete examples. Sample programs are provided on disk.

**Gain flexibility in designing your applications.**
CBTREE lets you use multiple keys, variable key lengths, concatenated keys, and any data record size and record length. You can customize the B+tree parameters using utilities provided.

**Your programs will be using the most efficient searching techniques.**
CBTREE provides the fastest keyed file access performance, with multiple indexes in a single file and crash recovery utilities. CBTREE is a full function implementation of the industry standard B+tree access method and is proven in applications since 1984.

**Access any record or group of records by:**
- Get first
- Get previous
- Get less than
- Get greater than
- Get sequential block
- Get all partial matches
- Insert key and record
- Delete key and record
- Change record location
- Get last
- Get next
- Get less than or equal
- Get greater than or equal
- Get partial key match
- Get all keys and locations
- Insert key
- Delete key

**Increase your implementation productivity.**
CBTREE is over 6,000 lines of tightly written, commented C source code. The driver module is only 20K and links into your programs.

**Port your applications to other machine environments.**
The C source code that you receive can be compiled on all popular C compilers for the IBM PC and also under Unix, Xenix, and AmigaDos! No royalties on your applications that use CBTREE. CBTREE supports multi-user and network applications.

CBTREE IS TROUBLE-FREE, BUT IF YOU NEED HELP WE PROVIDE FREE PHONE SUPPORT.
ONE CALL GETS YOU THE ANSWER TO ANY QUESTION!
CBTREE compares favorably with other software selling at 2,3 and 4 times our price.
**Sold on unconditional money-back guarantee.**
YOU PAY ONLY $99.00 - A MONEY-SAVING PRICE! NOW! Variable length records.
TO ORDER OR FOR ADDITIONAL INFORMATION
CALL (800) 346-8038 or (703) 847-1743
OR WRITE

PEACOCK SYSTEMS, INC

Peacock Systems, Inc., 2108-C Gallows Road, Vienna, VA 22180

by John Steiner

# Bug Bytes

## The Bugs & Upgrades Column

One of my major sources of information for bugs and upgrades is PeopleLink's AmigaZone. AmigaZone members provide a wealth of information about the Amiga, and events in the Amiga community.

I sometimes wonder if P-Link is worth the monthly bill of $25–$30, but last month one posting saved me at least that much in time. I don't remember who left the notice, but I wish to thank him. I saved several hours of hassle trying to transfer some IFF files to IBM format.

The A2088 Bridge Card DOS has a pair of commands called AREAD and AWRITE. These commands are designed to transfer files between the Amiga and IBM sides of the A2000. The commands work as promised in the manual with one exception: An undocumented option in AREAD and AWRITE writes a true binary copy of a file. This omission from the manual has caused much grief among bridge board users. To make an identical binary copy of a file, just append the command with /b. For example:

```
AREAD AMIGAFILE.IFF IBMFILE.IFF /B
```

Omitting the /b creates a file that is not an exact duplicate of the file being transferred. The /b option is not necessary when transferring text files, but should be used when transferring IFF or ARC files.

By the way, typing AREAD ? (or AWRITE ?) gives you a list of options, one of which is the /b.

Reports have persisted regarding a bug in early versions of Deluxe Paint II that causes a hard disk crash. Several users have reported the problem in versions of Deluxe Paint with a DH: button (instead of a DH0: button) in the load file requester. You can tell if you have the latest version by checking the buttons on the load file requester.

If your requester has a DH: button, and you have the unprotected version of Deluxe Paint II, you may request a free updated disk by sending your original. If you have a new hard disk, and still have the protected version of Deluxe Paint II, you can get the unprotected version of Deluxe Paint II by sending in the card that came with your original program, and $20.00. You will receive the latest version.

Send all upgrade requests to:
*Electronic Arts*
*Amiga Upgrades*
Box 7530
San Mateo, CA 94403

Electronic Arts requests no telephone upgrades. Send check, money order or Visa/MC credit card info, and allow 4–6 weeks delivery.

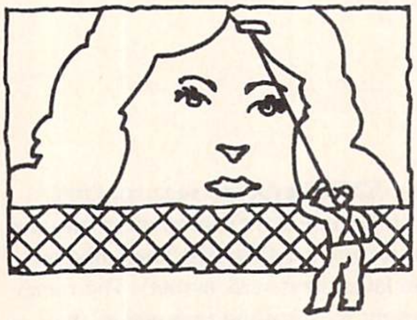
The A2000 keyboard saga continues with Commodore's announcement of a fix for the missing first keystroke on the latest keyboard model. The repair simply requires the removal of two capacitors on the A2000 motherboard. This alteration can be done under warranty by any Amiga authorized service center.

If you are brave enough to dive into your new A2000, capacitors C910 and C911 are the two that must be removed. These capacitors are two of several tiny blue capacitors just behind the keyboard connector. To get at them, remove the power supply/disk drive cage. They are located under the drive assemblies at the front of the computer. Remember, you will probably void the warranty if you make the modification yourself. Since the fix is free, you might consider having a Commodore service agency make the repair.

Commodore is currently shipping revision 4.3 of the Amiga 2000 motherboard. It no longer has the problem with the missing first keystroke.

On the topic of Amiga hardware, several reports have popped up concerning problems installing the 2010 internal floppy drive on the A2000. When the drive is installed, it acts like a df0:, rather than a df1:. In other words, accessing dh0: causes both drives to run. Accessing dh1: causes no response from either drive.

GET THE BIG PICTURE



"THE BIG PICTURE"

Print your IFF/HAM pictures in full color up to 104 feet x 104 feet. "Real" multiprocessing and overscan printed because printing is done from the file not the screen.

$29.95 for the following printers in one package:

Canon PJI080A / Okidata 20 color Radio Shack Quadran Quadjet / Epson FX-80, FX-100, EX-800, EX-1000 / Fujistu 1200, 2300, 2400 NEC P6, P7, CP6, CP7 / XEROX 4020

ATI-LINK
An IFF/HAM toTARGA file conversion program only $100.00

Perferences Drives
Quadran Quadjet $30.00 NEC CP7 $50.00

Big Picture is a trademark of Lightning Publishing. Amiga is a registered trademark of Commodore Ltd. Targa is a registered trademark of AT & T

**Lightning Publishing**
1821 N. Ohio St., Arlington VA 22205
(703) 534-8030

Name

Address

City State Zip

Amount Program

This strangeness occurs because, until recently, Commodore had been shipping all drives configured as unit df1:. They are now being shipped as df0:.

The solution is simple. Before installing the drive, check the drive select jumper setting. (In most cases, the jumper block is located on the bottom of the drive.) Most drive jumpers are numbered 0, 1, 2 and 3 or 1, 2, 3, and 4. If the jumper is on the end, move it to the next position in the row of four jumpers. If it is already located on the second row of pins, it is probably correct.

Several reports have surfaced regarding a "bug" in the keyboard software that causes the A2000 and A500 to ignore the + sign, parentheses, and some other number pad keys. This is not a bug; it is most likely caused by booting the A2000 or A500 on an old Workbench 1.2 disk.

A500 and A2000 keyboards must have the keymap set to USA1. A1000 computers require USA0, to which the operating system defaults. If you have an A500 or A2000, and the plus key on the number pad doesn't work, check your startup sequence file to make sure the command SETMAP USA1 is included. If not, add it to the file. The SETMAP command is found in the System subdirectory on the Workbench disk, so you will need to add a PATH ADD SYSTEM to your startup sequence if it isn't already there. You can also just RENAME SYSTEM/SETMAP AS C/SETMAP, then delete the SETMAP.info file from the system directory.

An upgrade is available for Transformer which allows it to run on Version 1.2 Kickstart/Workbench. (Officially, it runs on the 1000 and 500.) Commodore wants the 2000 PC support to come from the bridgecard. Such support is much more elegant, but also a more costly solution for those who need PC compatibility from their Amigas.

To obtain the upgrade, bring your master Transformer disk to your dealer. If your dealer is unaware of the upgrade policy, or does not have the disk, a call to his Commodore regional rep will get it to him.

The Spirit board internal clock software has a bug involving the leap year logic for setting system date and time. Even though the clocks were set correctly, the internal system date comes up one day ahead of the hardware clock. A call to Spirit disclosed that the Leap Year hardware and software conflicts until the end of February of leap year. A quick fix is to simply back-date the battery-clock, which will accordingly right itself at the end of February. Another fix involves modifying the startup sequence (until February 29, of course). After giving the command that reads the internal Sprirt Board clock, simply give a DATE TOMORROW command.

Mark Burton, (Plink ID, OER170) left me electronic mail regarding a problem with Dynamic Drums. When Dynamic Drums is booted up normally, the audio channels are full of noise. He discovered the program works properly only if the Mimetics Soundscape Sampler (attached to the second joystick port) is disconnected. If you have had problems with Dynamic Drums and the Soundscape Sampler, let me know.

That's all for this month. If you have any bugs or upgrades to report, you may contact me at:

John Steiner/Bug Bytes
c/o Amazing Computing
P.O. Box 869
Fall River, MA 02722

You can also find me as Publisher on Plink, or 73075,1735 on Compuserve.

•AC•

# The Ultimate Video Accessory

## *Part IV*

***by Larry White***

## Let's boot up and start making our desktop video commercial.

The screen is blank.

Slowly a colorful pattern appears, growing more intense until the entire screen is filled. Then the titles slide into position as the Amiga logo, its colors flowing proudly, centers itself above the words "Desktop Video." The color pattern wipes itself across the screen, behind the Amiga logo, which remains briefly as the entire screen fades to black.

This will make an exciting opening for our production, and we'll try a few of the most popular software packages as we go step-by-step through the process. We'll use Deluxe Video 1.2 (Electronic Arts) to choreograph all the action, so we'll have to be careful to keep all the pieces with proper resolution and palette for compatability.

The most complex part of this process will be capturing the Amiga logo and preparing it for animation by Deluxe Video (Dvideo). I found a clean color example of the logo on the box from my 1010 external drive, and cut it out so that it would be easier to handle.

To change something from printed art to a digital form that the Amiga can handle, you need a digitizer. Making Your Art Digital Digiview from NewTek is a digitizer which has become somewhat of a staple among Amiga based video systems. The basic system consists of a color filter wheel, a video interface for the Amiga's parallel port, and a software package. (Due to the differences in the pin configuration between the A1000 and the A500/2000 you'll need a different version of the hardware adapter, depending on which Amiga you're using).

You'll also need a video camera. You can use a color video camera or camcorder if you have one, but an inexpensive, high-resolution black and white camera (the type often sold for security surveillance) works best. You'll also need some lights and a tripod or copy stand to support the camera. The filter wheel shows the camera the separate red, green, and blue components of the object you're digitizing, and the Digiview program captures these three images separately, displaying the result in up to 4096 colors using the Amiga's HAM (Hold And Modify) mode.

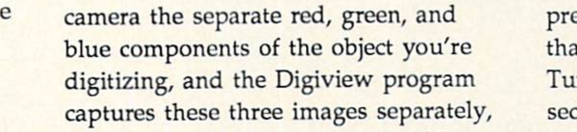


*Figure One*

Digiview software does not display the live video image, which makes it somewhat difficult to set up the camera. If you're using an Amiga monitor, you can plug the video line into the NTSC composite jack and switch the monitor to composite during setup. You can split the video signal using a Y connector, but this is still a bit awkward. A separate monitor is a better idea. I use an old black and white TV with an RF modulator.

Put the camera on the tripod or copystand with the filter wheel attached just in front of the lens. Place your artwork below the camera, preferably on a plain background, so that there are few extraneous colors. Turn the filter wheel to the clear section and turn on the lights, camera and monitor. If you're using a camera with manual lens control (such as the Panasonic CCTV camera sold by NewTek), open the lens iris wide, then focus. Adjust the camera to subject distance until the artwork fills most of the screen, but keep important details away from the edges. Now refocus the lens to get the sharpest possible picture, then gently close the lens iris until you find the point just before the image contrast looks flat.

Now boot the Digiview program. When the title screen appears, select 320X200 for compatability with Deluxe

(continued)

Video. Since we're digitizing for Dvideo, we can have only eight colors in the final image. Dvideo allows for two palettes of eight colors each, one for the background, one for the foreground and objects. There are several ways to achieve this: by image processing or conversion, or by letting Dpaint or Dvideo adjust the palette, but the best quality and most predictable result is obtained by setting the digitizer properly.

Select "Palette" from the control menu or press Amiga-P. A palette requester will appear. Adjust the colors slider to set the digitizer for a maximum of 8. Our logo will be a foreground object, so we'll need to keep color 0 transparent so that the background will show through. Click on Color 0 "Off"; this will limit the usable colors to seven. Make sure that the "Make New Palette" selection is highlighted, so that the computer will find the seven best colors to match your artwork. Then click on the word "color" to bring up the color requester. Set the mode to 8, and the dither to "off." Now click on the "OK" gadget to return to the main program.

Next, turn the filter wheel so that the camera looks through the red plastic filter. Select Digitize Red (or Amiga-r) to capture the red image. Then repeat for the blue (Amiga-b) and green (Amiga-g) images. From the control menu, select display. The color image will form in a few seconds. If the image looks good, you're ready to save it as an IFF file. If not, adjust the lighting and camera settings and try again.

If you plan to do a lot of digitizing, you might want to automate the process with NewTek's Digidroid, an Amiga-controlled motorized filter wheel which plugs into the second Amiga joystick port. With Digidroid, just select Digitize-Auto (Amiga-A) and the wheel will spin to each colored filter and digitize, then automatically switch to display the full color image. If you're already using Digiview 2.0, the Digidroid commands will be added to your menus when you boot Digiview with the Digidroid connected.

When you're satisfied with the image, select "Save" from the project menu . You should see the message "saving 8 colors" as the file is written to disk in IFF standard format.

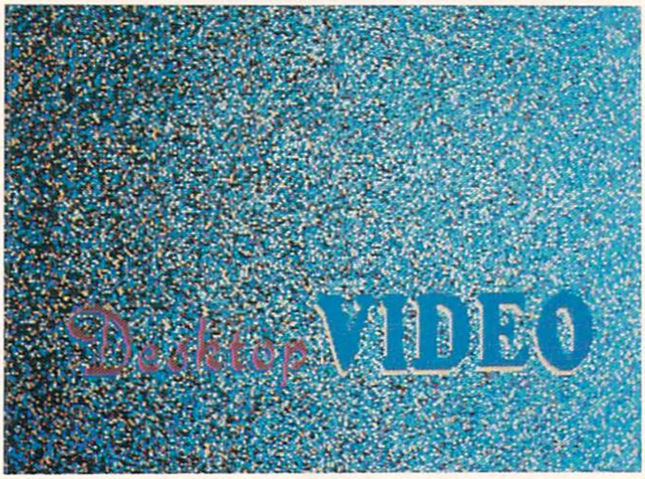


*Figure Two*

Now, let's use some image processing to prepare the image for animation.

### Digital Image Procressing

Two image processing programs are Butcher (Eagle Tree Software) and Pixmate (Progressive Peripherals). Although either could do the job, for this example I chose Pixmate. Pixmate is a total image processing system for the Amiga which can change pictures from one format to another, enhance contrast, edges, and grain, change colors, and even merge pictures.

If you look closely at the white background, you'll notice that it is really made of several different near white colors. Select "histograph" from the effects menu, and you'll see a bar chart which shows the number of pixels in the picture of each color. Three colors together represent white. Selecting the palette control from the Color Menu, adjust the sliders in the requester and change all three to the same white. Then, from the same menu, select "pack colors." This changes all pixels of the same color to the same selection from the palette . From the Info Menu select "display format" and you should see a message indicating that the image is low resolution, with 8 colors possible and 5 used.

You might use Pixmate to further enhance image details and edges, but that type of enhancement can be quite time consuming until you're very familiar with the software and all its features and settings. Let's save this file and use Deluxe Paint II to make some final adjustments.

### Color By The Numbers

We'll use Dpaint to create the other parts of our title and background, taking special care to select the palettes for use with Deluxe Video. Since we'll need large, clear letters, let's use Zuma fonts instead of the standard Amiga fonts. To do this, insert a Zuma font disk before loading Dpaint. Click on the Zuma disk and activate the font librarian. Select the fonts and sizes you want. Select "use" from the librarian, then exit from the librarian and start Dpaint II without rebooting the Amiga.

Select low resolution, 8 colors, from the Dpaint startup screen, then load the 5-color version of the logo. Although we'll actually be removing the white background and replacing it with the transparent color 0, it's easier if we save that step for last.

First, let's set the palette. Press "P" and the palette requester will appear. Color 0 is black, but since we'll need a opaque black in the letters, let's change color 0 so its location in the picture is obvious. Adjust the sliders so that color 0 is a color you don't want in the final image, such as violet. As you adjust the sliders, the colors in the image will change, making it apparent which pixels are controlled by which palette color.

To get the most effective palette, let's examine the specific image. We need black, white, gray, red, yellow, blue, and green. If we want the colors in the "checkmark" to cycle, they must be next to each other in the palette with no extra colors between them. This way, we can define them as a range (actually, this is true only in Dpaint, but not in Dvideo, where we can select each color and turn it on or off for cycling). Take one of the unused whites and change it to black. Use the fill gadget to fill the letters with this black anywhere you have color 0 (violet). Now, using the magnify gadget, you can fix any edges or shadows of the letters using the draw and line gadgets (figure 1). Make sure the outline of the checkmark is intact so that any fills you use will stay within bounds. Now use bright red, yellow, blue, and green to replace the colors in the checkmark. Fill the background and the centers of the "A"s with color zero to make the background transparent. Now use the brush gadget to make the logo a brush, then store it using the "save brush" command to conserve memory.

Now let's clear the screen and create the background. Use the control control to restore Dpaint's default palette. Set page size to full video so that we'll have a borderless screen for videotaping. Select a range of colors using the palette requester. Click the right mouse button on the fill gadget to bring up the fill requester. Adjust the dither slider towards the right to produce a fine mixed pattern, then select gradient for fill type. Return to the main screen and select the fill gadget. Press F10 to remove the title bar and menu from the screen, and fill the entire area with the dithered pattern. Save the full screen as an IFF file, then clear the screen.

Now, from the font menu select "load fonts." The fonts loaded will be those you specified from the Zuma Fonts disk. Once the fonts are loaded, select an appropiate font and a color which will contrast well with the previous background. Then select the text gadget, position the cursor, and type "VIDEO." Use the brush gadget to make the word a brush, select another color, and place the brush over the original word just slightly up and to the right, creating a drop shadow effect. Use the brush gadget to make the shadowed text a brush, then store the brush. Repeat these steps for the word "Desktop" using a different font and different colors (from the same palette!). Save this brush and exit Dpaint. We now have all the parts we need for our video scene.

**The Deluxe Way to Put It Together**
Run Deluxe Video. Dvideo is a program which loads these images and coordinates their movement on the screen. The program uses a timeline principle with tracks (horizontal lines) representing various images and effects representing specific movements and display commands indicating the start and end time (when appropriate).

Since we'll eventually be videotaping the video we're making, set the video for interlaced and overscan (to make the video borderless). Select "Restart" to make sure these settings will be in effect.

The video is composed of several scenes, but we'll just be creating one. Click on the scene gadget to bring up the "Scene Script." Then pull down a

(continued)

**I.R.S. No Sweat... With Money Mentor™!**

"Keep track of your pennies, and your dollars will take care of themselves." Old, but sensible advice, even in today's complex financial environment. Money Mentor™ is a breakthrough in personal financial management. It harnesses the awesome power of the Amiga™ to compute and graph clear reports of your financial situation.

A unique system called "Smart Scrolls" handles a diversity of tedious data entry functions and can save 70% of the typing typically required for entry.

Money Mentor™ features:

- 200 budget categories.
- 30 integrated accounts: checking, cash, saving and credit cards.
- Elaborate search routine allows editing of transactions according to your specific guidelines.
- Automatic check printing.
- Automatic Account Balancing.
- Colorful graphic reports illustrating actual versus budgeted amounts.
- Over 50 reports from which to choose.

**This year... get organized with Money Mentor™**

**SEDONA SOFTWARE**
11844 Rancho Benardo Rd., Ste. #20
San Diego, CA 92128
To order,
call (619) 451-0151

track box and place it along the left side of the script. When the requester asks you what type of track, click on "picture." Specify your background picture and select "get ready" from the load requester.

Pull down another track requester and select "background." Pull down an effect box and choose "fade in." The position of the left arrow of the effect box determines when the fade will begin. The distance between the two arrows determines how long the fade will take. The default is one second, but you can grab the second arrow and move it to make the fade shorter or longer. To see what this fade looks like, press Amiga-k.

Pull down another track box and select "object." Load the brush containing the word "VIDEO" as this object. Put an effect box at the begining of the object track. When the effect requester comes up, select "appear." Another requester will pop up labeled "appear where?" You can respond three ways: by entering coordinates, by moving a representation of the object to the position on the scale screen, or by clicking on the TV set, which will produce your object and the video screen. (If you've already played the scene, the background should be visible). You can now grab your object and position it exactly where you want it to appear. I put the "VIDEO" title off screen at top left.

Bring down another effect box, placing it just after the "appear" effect. This is a "move to" effect. Again the TV gadget will show you the Dvideo screen and you can put the object exactly where you want it. Put it below and to the right of the center. You can play the scene to check the movement, clicking on the "move to" box if you want to change it. If you want to adjust the timing, click on the second arrow and move it.

Pull down another track and select "object." Now load the "Desktop" brush. Using the same technique as the "VIDEO" brush, make "Desktop" appear from the left side,(after the "VIDEO" object starts moving). Then move it towards the right so it stops at the same time as "VIDEO", spelling "Desktop VIDEO." By trial and error you should be able to make it appear as if the word "desktop" pushed the word "VIDEO" to the right.

We're almost ready for the Amiga logo. Remember, the logo has a different palette than the other two objects. If we were to introduce it suddenly, the other objects would change their colors because of the new palette. What we need to do is stamp the objects, copy them into the background, (which we created with the same palette), then make the objects disappear before we introduce a new object and its palette.

Make the Amiga logo appear centered, just off the top of the screen, immediately after the other objects dissapear. Then move the logo into position just above the words "Desktop Video." Add another track to the scene, this time a foreground track. Pull down an effect box and position it so it starts simultaneously with the appearance of the Amiga logo. Make this a "cycle colors" effect, selecting the colors in the checkmark to cycle (I used 1–4). Bring the end of the effect to the end of the scene. Press Amiga-k to play the scene and watch what happens!

I added a few touches to end this scene smoothly. First, I wiped the background from left to right, replacing the colored background but leaving the Amiga logo. Then I faded the foreground and background to black, ending the scene (See figure 2).

Of course, there's a lot more to Deluxe Video including the ability to animate an object as it moves, size an object, or even use a special text which can be rotated 360 degrees along three axes. Add music tracks and digital sound effects and you'll have an even better impression of this powerful and fun program. I recently had the opportunity to attend a demonstration of two soon to be released programs from Electronic arts which will also suppliment Deluxe Video: Deluxe Productions (which I have previously referred to as Deluxe Titler) and Deluxe Photolab, a combination HAM paint program and image processor.

Next time we'll continue our video using the Director—a programming language similar to Amiga basic, but with a number of video control commands.

•AC•

## Suppliers:

*Zuma Fonts*
**Brown-Wagh Associates**
16795 Lark Ave.
Suite 210
Los Gatos, CA 95030

*Butcher*
**Eagle Tree Software**
PO Box 164
Hopwell, VA 23860

*Deluxe Video, Deluxe Paint II*
**Electronic Arts**
1820 Gateway Dr.
San Mateo, CA 94404

*Digiview, Digipaint, Digidroid*
**New Tek**
115 West Crane St.
Topeka, KS 66603

*Pixmate*
**Progressive Peripherals & Software**
464 Kalamath St.
Denver, CO 80204

*The Director*
**Right Answers Group**
Box 3699
Torrance, CA 90510

# The *Hidden Power* Of CLI Batch File Processing

***By Joe Rothman***

Amiga Mouse Users Group A.M.U.G.
BBS 516-234-6046

Icons are great, and the Amiga's four-color Workbench is really the best I've seen. Remember the first time you used your Amiga? What power! The Workbench, however, is only the surface; the Command Line Interface (CLI) has even more power. CLI allows you to see files and directories that don't appear on Workbench.

To varying degrees, we all use CLI. Some of us just use a few commands like Dir, List, Copy, Format, and Diskcopy. Some of us have set up virtual drives such as ASDG's Recoverable RAM Disk, or collected CLI Shells and dozens of PD commands and programs which we access through CLI. Even if you very rarely use CLI, knowing how Amiga Batch files work can make your Amiga easier to use.

A Batch file is basically an ASCII text file; the Execute command interprets each line as if it were a command typed on the CLI. Every time you boot up the machine, the Amiga uses a Batch file called the startup-sequence. That file is stored in a disk directory called the s directory. The s directory has a very special property under AmigaDOS: every time someone uses the Execute command, AmigaDOS automatically looks for the filename after the word Execute in the s directory of the current Workbench Disk. It follows, then, that all Batch files should be stored in the s directory of the Workbench disk with which you intend to use the files.

Before I show you some sample Batch files, I would like to present a little tutorial on ED. You use ED, which is a DOS command, to create Batch files. (Batch files can also be created using any word processor or text editor.) Everyone has access to ED. You can create a new Batch file with ED by typing Run ED SYS:s/filename. The ED screen appears with a cursor at the top and the message "Creating New File" at the bottom. If you use ED on an existing file, the ED screen has the existing text in the window when it opens.

> Even if you very rarely use CLI, knowing how Amiga Batch files work can make your Amiga easier to use.

There are a few things you must know to use ED. When you enter a line of text into ED, you must hit return at the end of each line, because there is no word wrap. There is also no mouse support, so use the cursor keys to move things around; use the Backspace and Delete keys to edit the text. Press ESC, Q, and Return to quit without saving; press ESC, X, and Return to save and quit. The bottom of the ED screen is the command line, where special messages appear.

### Format Batch File

Let's try our first Batch file. Here are the ground rules for copying the examples in this article: upper and lower case letters are not important, but spacing and spelling must be exact.

### Batch Disk Formatting

Everyone formats disks, and there are several ways to do it. Workbench allows you to click on the disk and select Initialize from the disk menu. That's easy, but if you want the disk to be named anything other than Empty, you will have to use the Rename command on the Workbench menu. And if you don't want the trashcan, you have to delete it from CLI. Obviously, you don't have a lot of control when you use Workbench to format disks.

On the other hand, CLI allows you to control everything, but makes you type a very long command. Let's say I want a disk to be formatted in drive DF0: without the Trashcan, and I want it to be called PDGames. Here is the CLI command to do that:

```
Format Drive DF0: Name PDGames NoIcons
```

That's a lot to type just to format one disk, and if you are not using a CLI Shell, you will have to retype it every

*(continued)*

time you want to format a disk. You could, however, make up a Batch file to format disks for you. Create a file called Format stored in the s/directory of your Workbench disk. Since you intend to use the file to format many disks, the name you may decide to use is "Empty." The Batch file must consist of the following:

```
Format Drive DF0: Name Empty NoIcons
```

There are three problems with using a Format Batch file like the one above: 1) it always makes you format the disk in the DF0: drive; 2) it always names the disk Empty; and 3) it returns an error message from DOS, "Format failed returncode 20," even though the disk is formatted properly. The error message comes up because of a minor difference in the way AmigaDOS treats Batch files, as opposed to how commands are actually entered at the CLI prompt. (I don't know exactly why it happens, but I do know what to do about it.)

Let's solve those three problems. We can make our Batch file format the disk in any drive, even a hard drive; we can make it demand that we give it a drive to format the disk by having it give us an error message if we don't; we can make it accept a different disk name from CLI, and call the disk Empty if no name is given. It sounds as if we will need a huge Batch file, but these "features" require only a three line Batch file:

```
.key Drive/a,Name
Failat 25
Format Drive <Drive> Name
        <Name$Empty> NoIcons
```

The .key statement must always be the first statement in the Batch file. Just keep that in mind if you decide to add anything to this file, or to use a .key statement in another Batch file. Drive/a requires that parameter <Drive> always be present, or you will get an error message. Name is optional, and <Name$Empty> establishes Empty as the default name to use if none is supplied. Failat 25 supresses the phony "Format failed" error message I mentioned before. If you prefer to have the trashcan on your disks, just leave the word NoIcons out of the Batch file.

Assuming that you called the Batch file "Format" and stored it in the s directory of your current Workbench disk, here is how to use it from CLI:

Enter the command "Execute Format DF1: PDGames [Return]" to format a disk in DF1: with the name PDGames.

Enter the command "Execute Format DF0: [Return]" to format a disk in DF0: with the default name Empty.

As you can see, the Batch file called Format has given you as much versatility as typing the actual DOS command, without a lot of typing. By using the assign command, you can shorten the command even further. At the CLI prompt, enter "Assign x: sys:c/Execute [Return]." Every time you type x:, the Amiga replaces it with sys:c/Execute; now all you have to type is "x: Format DF1: PDGames [Return]" to format a disk in DF1: with the name PDGames.

## Batch Alarm Clock

The Amiga comes with a clock that has a built-in alarm. The alarm clock is not very good; it is hard to set and makes the screen flash only once at the specified time. If you happen to turn away at that moment, you might miss the alarm completely. Try this Batch file:

```
.key Time/a,Verb,Noun
.def Verb "Do"
.def Noun "Something"
Wait until <Time>
Echo "It's time to <Verb> <Noun>"
Say "It's time to <Verb> <Noun>"
```

To test it, enter the following command at the CLI prompt (substitute an actual time in 24 hour format for the HH:MM).

```
Run Execute Alarm HH:MM (Return)
```

The Batch alarm works best if you are running your system from RAM: or VD0: because the Echo and Say commands must be loaded from disk when the alarm goes off. The preceding example uses the defaults established in the .def statements and statements, and your Amiga waits until the specified time and then sends this message to the screen, "It's time to Do Something." When you set the alarm, you can replace the <Verb> and <Noun> defaults with anything you want. The replacement text can be one word or a string of words. (Use quotes to enclose the string so Amiga DOS knows what to do about the spaces.) Here are two examples:

```
Run Execute Alarm HH:MM Watch TV

Run Execute Alarm HH:MM "Pick up"
    "the Kids"
```

Because the Wait command will tie up its CLI until the specified time, you must "Run Execute" instead of just "Execute" the Batch file alarm clock. The Batch alarm writes a temporary file to the t directory of the disk you are CDed to if you don't want the temporary file written to your disk CD to RAM: before using the alarm. For some reason, CD will not work between the .key statement and the wait statement. You don't have to worry about whether or not the t directory exists, because if it doesn't, it is automatically created. If you wish to check whether the alarm task is active in the background, you can use Status [Return] at the CLI prompt, and you will see one of the tasks listed as "Wait." If Status says that the Wait command is active in task 4, you can cancel it by using the command Break 4 [Return].

You can modify the Batch alarm to use the PD Beep command, instead of the Echo or Say commands. For that matter, anything can be tacked on after the wait command, allowing you to use it for fully-automated and unattended tasks. For example, you could have it run a terminal program which uses a script file to call an on-line service and captures some information at a specified time without any assistance!

**Dirall**

This next Batch file will allow you to print an entire disk directory listing all the files in all the subdirectories on the target disk. Call this one "Dirall." The Drive specifier is mandatory, but the <Dest> parameter will default to PRT: if no Drive:Directory/Filename is specified.

```
.key Drive/a,Dest Dir > <Dest$PRT:> <Drive>
    Opt a
```

You can type either of these commands:

```
Execute Dirall df0: [Return]

Execute Dirall df0: Ram:DirZero (Return)
Type Ram:DirZero (Return)
```

**DCopy**

Let's make a universal Diskcopy Batch file that will work on one and two drive systems. Call it DCopy and store it in the s directory. (Please be careful when you try this example; don't copy over a disk you need.)

```
.key From/a,To/a
Diskcopy DF<From>: to DF<To>:
```

Now try using DCopy with these commands:

```
Execute DCopy 0 0
```

or

```
Execute DCopy 0 1
```

You can use the .key to create an argument template and .def statement to assign a default parameter in Batch files of all kinds. If you forget which parameters you are supposed to enter at the CLI to activate one of these special Batch files, all you have to do is enter "Execute filename" (substitute the name of the Batch file for "filename"). AmigaDOS responds by typing out an error message which names the parameters.

Experiment—see if you can create your own useful Batch files, using the parameter substitution abilities of the Execute command, and pass them on to other Amiga users on BBSs and Networks throughout the land.

•AC•

GOMF!

"...it is small, installs with just a double-click on an icon...a welcome utility for Amiga users."

-INFO magazine

"...Without a doubt, the most important program this month is GOMF. All Amiga users are sick and tired of losing running programs and data to the dreaded Guru, and anything that can stop this terrible syndrome deserves a round of applause!"

"...So if Gurus are getting to you, add GOMF to your Amiga environment and you can start saying goodbye to the GURU too!"

-Commodore magazine

Now GOMF is even better!
Version 2.0 NOW shipping

NEW FEATURES include:
- Automatic removal of ALL errors (task held, GURU, etc.)
- Preserves low memory auto vectors.
- NUKE those pesky CANCEL requestors once and for ALL!
- Eliminate ANY task or device AT ANY TIME.
- auto-configures for 68000/68010/68020.
- CLI and Workbench compatible.
- Extensive, easy to understand documentation.

only $39.95

New Products for All *AMIGAS*
from HyperTek/Silicon Springs

Deluxe MIDI interface

Simply the BEST full-featured MIDI interface available for the AMIGA 500/1000. Fully compatible with all programs that use serial MIDI standard output, ONLY the DELUXE MIDI INTERFACE offers you: SERIAL PORT Pass-thru, MIDI IN, MIDI THRU, and two SWITCHABLE THRU/OUTS ! This is the optimum configuration for anyone with more than one synth or other MIDI device. No THRU on your synth(s)? No problem! Simply flip a switch on the Deluxe MIDI interface, and avoid costly THRU boxes! Includes 8 foot cable, 1 year warranty.

Please Specify model 500/1000..........$89

TTL Hi-res Monitor Adapter

For the Amiga 500/1000/2000. Plugs into RGBI port for ULTRACRISP flicker-free high resolution monochrome video output. Perfect for HI-RES GRAPHICS, WORDPROCESSING, CAD, DESKTOP PUBLISHING, BUSINESS, etc. ANY application in ANY resolution is SHARPER and CLEARER with the TTL Hi-res Monitor Adapter. Perfect for use with the low-cost (under $100) Commodore 1901 monitor or Monitor 80. Includes easy installation instructions, a disk with a special Hi-Res WorkBench font and RGB port pass-thru (a 1080 RGB monitor may be used simultaneously). 1 year warranty.

*Be KIND to your EYES!*

(Monitor not included)..........$99.95

Light Pen

Designed to work with ANY Amiga program, the LIGHT PEN and DRIVER allows use of both your mouse and a pro-quality LIGHT PEN for the ultimate in precision graphics. Perfect for PAINTING, DRAWING, freehand SKETCHING, CAD and virtually ALL other AMIGA programs. Software features include: Single-pixel precision, variable sensitivity, ZOOM mode, button toggle, etc. Includes Inkwell Systems' DT-184A high quality two button lightpen. *MAIL ORDER SPECIAL:* While supplies last, we'll also include a handy PEN-HOOK that attaches easily to your monitor to hold the AMIGA LIGHTPEN when not in use!

(500/1000/2000)..........$129.95

All products are NOW SHIPPING! Please make cheque or money order payable to:

HyperTek/Silicon Springs
#120-1140 Austin Ave. Coquitlam, BC Canada V3K 3P5
Phone (604) 942-4577 Dealer Inquiries invited.

All orders add $4 shipping. All prices in US$
Order by phone! VISA, Mastercard, AMEX welcome.
U.S. orders add $3 handling fee.
AMIGA is a registered trademark of Commodore-Amiga, Inc.

# A Conference with Eric Graham

***Edited by John Foust***

This is a transcript of an online, electronic conference with Eric Graham, author of the Amiga modeling and animation programs Sculpt 3D and the upcoming Animate 3D.

The conference took place on November 18, 1987, in the Amigazone's weekly Wednesday night Graphics & Video slot. Other conference guests included Ken Offer of Byte by Byte, who had two winning entries in the BADGE Killer Demo contest.

The first portion of this transcript includes the opening remarks and a moderated question-and-answer portion. Eric Graham uses the handle **(*ERICG)** and Ken Offer is **(*KEN)**. The comments of other participants are prefaced with their handles.

**(CBM*HARV)** Welcome to our Wednesday night graphics and video conference. Tonight we have something special for you. Our special guests are Eric Graham, author of Sculpt 3D and creator of the "Juggler" animation, as well as Ken Offer, the creator of "Kahnankas" and other incredible Sculpt 3D animations. Eric, let's start with you. How did you come to the Amiga? What is your general background?

**(*ERICG)** OK. I wrote a ray tracing program in 1966, but there was no good computer to run it on. The only machine (the biggest in Europe) took many hours to make an image, and I had to print it out on a teletype. 80 by 66 pixels! I got the Amiga a year ago. It was the first machine with good enough graphics to do justice to ray tracing. "The Juggler" was produced in about a month, and on December 26th, I started writing Sculpt 3D. The difficult part was the interactive interface. It was finished in August, and because I immediately started on Animate 3D, I haven't even had enough time to play with Sculpt. Ken has had all the fun!

**(*KEN)** Sure! And I love every minute of it!

**(CBM*HARV)** Thanks Eric.

**(CBM*HARV)** Ken Offer? Do you care to tell us a little bit about yourself and your Amiga work?

**(*KEN)** Well, I am also one of your first Amiga owners, crazy about this machine from the start. My main interest in computers is my first love: art.

**(CBM*HARV)** Ken, can you name your Amiga animations for folks here who might not know which titles you created?

**(*KEN)** I work for Byte by Byte, so my opinions are totally my own. My animations are the "Kahnankas," the "RotAmiga," the "BoingMachine." I have done the sound upgrade for Movie, and new versions of all the animations currently exist.

**(CBM*HARV)** We'll be uploading "BoingMachine" into the Club library tonight. Thanks, Ken.

**(*KEN)** Great.

**(CBM*HARV)** Okay folks, any questions for our guests? Let's take them one at a time, please.

**(CBM*STEVE)** Eric, have you considered adding texture mapping for surfaces in Sculpt 3D? As in having an IFF picture defined and telling which object you'd like it mapped to?

**(*ERICG)** Yes, we should have texture mapping available soon. The principal difficulty is the user interface. How do you wrap King Tut around an elephant and know what to expect? We'll have a good interface soon.

**(CBM*STEVE)** Heh, heh. Understood! Thanks!

**(JRH)** Eric, are you using the ANIM standard or adding to the standard for Animate 3D?

**(*ERICG)** Yes, we use the ANIM standard, but have our own op-code. It will be made public as soon as it is frozen.

**(GSARFF)** I was just wondering about the Movie utility. Why does the disk stay spinning and the light on when the animation is running?

**(*ERICG)** It seems to be a bug in the Manx C compiler overlay system. It doesn't do it on a hard disk. I will try to find a workaround for it.

**(ROGER*B)** Mr. Graham, will Sculpt 3D and Animate 3D support the MC68020 and 68881?

# on People Link's AmigaZone

**(*ERICG)** Yes and no. The standard version of Sculpt uses the fast floating point library functions for speed, but we have a special version that uses the '020 and '881, and it is FAST. There will also be a version for Animate.

**(SWALLOW)** Could you list some specifications on Animate 3D as compared to Silver and the current version of Movie?

**(*ERICG)** Animate is fully interactive. You use the mouse to design motion paths which can be hierarchical (wheels within wheels). Also, you can perform key frame animation. Unlike Movie, the whole process of generating an animation is done with one mouse click. No script files needed! It also has all the features of Sculpt 3D so you can design really complicated objects, not just spheres and other simple shapes.

**(CJCARTER)** Are there any plans to make an ANIM player that will allow user input?

**(*ERICG)** Not at this time; we seem to fill all available time, so there isn't enough to allow the possibility of much useful interaction with a movie.

**(DJJAMES)** Ken, where did you come up with the name "Kahnankas?" I have a hard time spelling it.

**(*KEN)** Well, I really did not know what those things were called. I used to have a set as a kid. So I came up with a name that sounded like what they sound like. By the way, we have a new sound version of it that clicks.

**(CBM*STEVE)** Ken, were those animations designed using Sculpt 3D and Movie, or did you use a prerelease version of Animate 3D?

**(*KEN)** Yes, I made them totally through Sculpt 3D. I animated them through the script command system.

**(ROGER*B)** Eric, is the blitter used much, and how much memory is required?

**(*ERICG)** The blitter is used mostly for fast update of the Tri-View. Both Sculpt 3D and Animate 3D will take advantage of all the memory you have. Sculpt 3D will work on 512K, but really needs 1 meg. Animate NEEDS 1 meg.

**(SWALLOW)** Ken, what is the "sound" upgrade for Movie, and has it been made available yet?

**(*KEN)** The addition of sound is done in a similar manner to the PILBM command file. An additional parameter is given to Movie to specify this file. The file contains a list of IFF sounds, and a description of how to play those sounds. Foreground and background sounds are defineable. The new version of Movie is called "NewMovie" and is available on the "Sculpt_Anim1_w-sound" and "Sculpt_Anim3_w-sound" disks, just released into the public domain.

**(WALRUS)** Where can I get a copy of "Kahnankas" that clicks? That's the only thing that I thought was missing from a fantastic demo.

**(*KEN)** We are sending the disk to Harv. Also, I will be passing it out to members of our local user group and probably uploading it to the nets, along with documentation.

**(AMICUS)** I have it, so I can upload it...

**(CBM*STEVE)** First, thanks to both of you for being here! I love Sculpt 3D! Here's the question: Eric, how do Sculpt 3D and Animate 3D interact? Or do they? Is Animate 3D a standalone package?

**(*ERICG)** Animate 3D requires Sculpt 3D. The first thing that you do is merge them together. Then you have one (big) program that combines all of Sculpt's features with the ability to interactively design animations.

**(CBM*HARV)** Eric, can you explain what "merge together" means?

**(DJJAMES)** Merge them together—what a great idea.

**(*ERICG)** Both programs consist of lots of cooperating modules, so the merge is easy.

**(DJJAMES)** Eric: I'm not familiar with J compression. Can you explain it briefly? If not, just say so.

**(*ERICG)** It is an adaptive system that looks at the images and chooses compression schemes that suit the data. That way, you can squeeze a bit more compression and a bit more speed on playback.

*(continued)*

**(OJSANDS)** Eric, are there constraints on the length of an ANIM animation? How long is possible?

**(*ERICG)** It is only limited by the amount of RAM. Currently, if you use floppy disks, that limits you to 880K. We must make it possible to do multi-floppy ANIMs.

**(ROGER*B)** Mr. Graham, how many bitplanes are possible, and what is next after Animate 3D?

**(*ERICG)** HAM uses 6 bitplanes; that is all the Amiga can handle. More bells and whistles? What do you want? Texture mapping is on the top of the list.

**(*KEN)** Sculpt 3D can do 24 bitplanes, by the way.

**(WALRUS)** Will multi-floppy ANIMs be implemented any time soon?

**(*ERICG)** Let me know if you are ready for them! Not difficult, just give me a day of free time.

**(*KEN)** Marcus (a co-worker of mine) and I are thinking about how to link animations.

**(AMICUS)** I think the splitting of animation files is an important problem that Aegis hasn't really addressed.

**(AMIGA*GREG)** How long did it take to generate "Kahnankas" on a 68020?

**(*KEN)** About "Kahnankas"—about six hours per frame and a total of 16 frames on a 68881/68020 without 32 bit memory.

**(CBM*HARV)** Eric, do you have a release date for Animate 3D yet? And what price?

**(*ERICG)** Real soon now. Animate 3D will list at $149.95.

**(AMIGA*GREG)** How in the world was the banker lamp made in "Kahnankas"? And how long?

**(*KEN)** The lamp was easy. A couple of mirror cylinders and a spin or two.

**(CBM*STEVE)** Ken, any general tips on animation for us novice animators?

**(AMICUS)** Ken, I'd love to hear more about your techniques for scripting animations for Sculpt 3D sometime ... even if Animate is coming soon.

**(*KEN)** General tips ... Well, I can name quite a few, and maybe I should write an article on it. It is a big subject. A little long for this conference.

*This entire transcript is Copyright (c) 1987, the People/Link AmigaZone and may be reprinted in part or in whole only upon receipt of express written permission to do so. Contact CBM*HARV on People Link for more information.*

**TxEd *Plus...*** All the speed and simplicity of the original TxEd, *Plus*...

Modular software is coming. With modular software, different programs talk to each other using a common macro language. Apple Computer, Inc. has started a software division focusing on modular software, and Microsoft Inc's Bill Gates has been talking about it. Modular software lets "multitasking" mean more than just running two programs at the same time; you can run programs *together*, doing more than each program can do alone and letting you pick exactly the pieces you want to use.

On the Amiga, modular applications are more than just next year's dream. They're available now, with two of the cornerstones ready to go. TxEd *Plus*, the text editor, and AREXX, the macro processor.

Even without the AREXX connection, TxEd *Plus* is the text editor of choice for the Amiga. With AREXX, TxEd *Plus* becomes more than just a text editor. The configurable menus allow you to create customized applications such as order entry systems, and that's barely scratching the surface of the possibilities. We don't even know what the limits are yet.

Developers: find out about AREXX! The window is wide open. Users: demand AREXX capability in your software. You can get it now on your Amiga, or wait till next year, on a McClone.

○ Fully configurable menus and keyboard
○ Powerful command line language, uses the AREXX macro processor.
○ Includes functional AREXX demo
○ Many other new features.
○ Uses ARP V1.1

**TxEd *Plus* $79.95**

TxEd

**Microsmiths, Inc**

PO Box 561, Cambridge, MA 02140
(617) 354-1224
BIX: cheath CIS: 74216,2117

Mass Residents add 5%. Visa & MC accepted.

Amiga is a trademark of Commodore Business Machines, Inc.

The term 'McClone' is a fictitious conglomeration refering to nothing in particular.

# Amazing Computing *Catches up with* Perry Kivolowitz, President of ASDG Ltd.

### *by Ed Bercovitz*

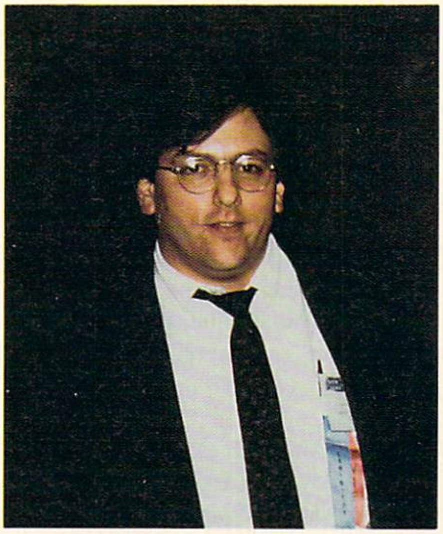
Perry Kivolowitz has been a major figure in Amiga user and developer communities since the earliest days of the machine. His company designed memory boards and card cages for the A1000. In early 1988, ASDG will release two new, innovative products.

The first, the 2001, is an expansion box for the Amiga 1000 (pre-summer release.) It will permit nearly full-functional equivalency to the 2000, with expansion slots for old and new Zorro cards, as well as IBM-AT standard cards, a co-processor slot, space for internal hard and floppy drives, and a 200-watt power supply.

The second product, planned for release in April, is the Satellite Disk Processor (SDP). This intelligent controller will support both SCSI and ST-506 drives. It will contain an onboard 68000 processor with 512K of cache RAM, and promises a multi-fold increase in hard disk input/output over any other Amiga hard disk currently on the market. AC talked with Perry about these products, other software products, and the early days and future of the Amiga.

**AC:** *Perry, with your articles in* Amazing Computing, *your active participation in major electronic networks like Usenet, CompuServe, and Plink, and your presentations to various users groups and at shows like AmiExpo, you've become rather a well-known figure in the Amiga community. How do you feel about this fame?*

**PK:** All of this notoriety has just popped up on me. Favorable review by peers is an extremely strong motivator. I was very heavily involved in the Unix environment—publishing papers and attending Unix conferences—and never managed to get any closer to the inner circle than the outer fringes. It was very frustrating because I spent years working on it, while overnight I became a well-known personality in the Amiga world. That was a very strange adjustment. The novelty, of course, has passed, and I am still me. What gives me a great amount of pleasure is when people come and say, "You did a good job, Perry." That has given me a tremendous feeling, both personally and professionally.



**I don't think manufacturers should be in business strictly to make money if it is done at the expense of providing service. If you can't do both together, you shouldn't be in business.**

**AC:** *I'm always fascinated by people's backgrounds before becoming involved with the Amiga. What was your experience?*

**PK:** I started with computers when I was about 10. There was a guy, who was nine years older than me and a friend of my older brother, who, when he was in high school, would be taking apart IBM machines and

(continued)

putting them back together differently, but better than they were originally. He introduced me to the world of computers.

So from 10 years old that was the thing for me. I started reading all the books I could find. I didn't have a computer or even know anyone who had a computer. I had to wait until I got to high school until I could even get near a computer. And that wasn't even at my high school; I had to take a bus across town to another school that had an IBM 1130. It had an amazing 4K of memory. I had to beg and bribe the other students to get any time at all on the computer. We did things like writing programs that moved hard disk heads around, thereby resulting in various types of radio frequency interference which would cause varying sounds to be emitted from an AM radio. By controlling the movements and thus the sounds, we were able to produce our own "songs."

When I went to university, I found the undergraduate program lousy. The staff there basically considered me a hacker. I was a necessary evil because if hackers didn't exist none of the projects for their grants would get done. But to them, a hacker wasn't a theoretician or an academic.

Graduate school was where I learned why everything works the way it does; it was where I got my theoretical foundation. Once you know the theory, it allows you a repertoire of solutions broader than any hacker could have in his lifetime. A theorist by himself is useless, a hacker by himself is of limited usefulness, but put the two together and they can't be stopped. Hacking is great. You can't have a computer industry without hacking, but you can't have a computer art and science without theory.

So after graduate school, I wanted to get a steady job, so I went off to Wisconsin and worked for a company there as the Unix software products manager. Unfortunately, there were no Unix products, so I got to start the department. I should have known there was trouble coming when they said, "Before you start the department, there's a few things we want you to do." It turns out that they had sold software products two years before, but had never delivered. So I spent nine months doing those products to get rid of their backlog. Then I started to put together their department. It turns out that they weren't really very interested in having a Unix software department. So I left. There was a parting of ways, and I came back to New Jersey.

**AC:** *So this was when you started getting involved with the Amiga?*

**PK:** Yes. After returning from Wisconsin, I was getting real "antsy." I was about to buy a Macintosh because I was saying, "I'm getting into business for myself. I hate working for other people. I'm going to start writing software. I can do this just as well as anybody else."

Then I got a call from a friend (Rick Sperenbauer who later designed the Ameristar Ethernet board) who said, "Did you ever hear about an Amiga?" So I started getting magazines and became the enemy of everybody I knew for a few weeks because every time I saw somebody with a magazine that had something to do with the Amiga, I would make it disappear. I needed the information. I was frenzied.

I got my Amiga in October 1985 on the first day they were available. I drove from New Jersey to the Eastern tip of Long Island and paid full list price (and to find out there was another store selling them the same day not more than 2 miles from my house).

I got half a meg of RAM, one disk drive, but I had to use my TV, instead of a monitor. I had to sit with the whole Amiga in my lap because my TV was one of these big jobs and I had no desk at home. All we had was Boing, Kaleidoscope, and Robocity to watch. You would be surprised [for] how many hours we were happy watching those.

At that time, I was always on Usenet as a Unix hacker, and, at that time, I put out the first public domain program for the Amiga on a major public network. This was the first program I wrote for the Amiga. It was a 3D animation of one of those kinetic desktop toys with silvered balls bouncing off each other in a pendulum frame. *[Editor's note: See "Balls" on Fred Fish Disk #1.]*

There will always be a romantic attachment to the early days of the Amiga. Eric Lavitsky, my partner and Head of Technical Support Director at ASDG, is one of the most unsung heroes of the early days of the Amiga. He started the Amiga Usenet group. He and I went to Connecticut, New York, New Jersey, Delaware, and Pennsylvania—we went everywhere showing Amigas. We were among the few people who had Amigas and knew what it was. We went to Amiga groups all over the place showing people the Amiga and making the early converts.

**AC:** *You have developed a reputation as always being willing to help anyone, from a beginner to someone who is very experienced and is having a problem. Do you have a particular philosophy in this regard?*

**PK:** Conveying information to other people, and having them learn it and understand it, is as satisfying as having someone come up to you and shake your hand and say, "That was a great program that you wrote." It turns out that a lot of people are afraid of computers and they don't understand them.

Computer programming is really more like learning to navigate. If you're good at giving somebody directions to get from one place to another that they've never been before, then you can be a computer programmer. Because after all, what is the computer? The computer is a naive traveller, and you set it upon a course; you're its navigator. If your directions are good, it gets where you want it to go.

People ask me, "How can I learn about computers?" I'll tell you how I learned. I had an interest in Unix and a curiosity, so I kept asking questions. Luckily, there was a person around who was a Unix guru already. He was great. His philosophy and attitude towards people who had questions was to give everyone a budget of dumb or repetitive questions. If you were under budget, he would be real glad to answer you, even if you asked the same dumb question five times. But as you started using up your budget, he would give you less and less information, forcing you, if you were really interested, to find the answer out on your own.

So after a while I used up my budget of dumb questions, and I was given a stack of source code about an inch high and was told, "All the answers are in here. If you have any more questions, look in here." So I took the source code home and read it for days and days and days and weeks and months till I learned it.

So how do you learn something? Find a guru and use up your budget of dumb questions. After that, it's up to you. You've got to take the initiative to find the information that's out there. It's up to you to get it.

**AC:** *One of your most well-known products is VDO:, the Recoverable RAM Disk. Could you tell us a bit about its development?*

**PK:** People tell me I don't write code like other people do. For example, the Recoverable RAM Disk I started Thursday night and finished Friday night. It would have worked the first time except every time I opened it directly, it worked great, while if I let AmigaDOS open it, it crashed. I was tearing my hair out and ended up not sleeping from Thursday to Sunday. I started looking at my source code and finally decided the problem wasn't there.

Then I started looking at the assembly language output of the compiler. I was using a really early version of Manx. I saw an instruction that was off by one bit. I changed it in the assembly language and it worked!

The next day I was so mad, I called up Jim Goodnow of Manx at home, and I started screaming at him about what happened. His response was, "Oh yeah, I know about that problem. That was fixed weeks ago. You shouldn't be using that version of the compiler for developing." And, of course, he was right. Even so, I was still so mad that I yelled at him, "Thanks a lot and have a good day," and hung up. This was in September, 1986.

The reason I wrote the Recoverable RAM Disk wasn't my idea. I heard a rumor that another hardware company was doing one, and I said to myself, "Uh oh, we're going to go out of business unless we have one." So I wrote it, but the company who I heard the rumor about didn't get their RRD out until mid-1987. Originally, it was proprietary (i.e., we only gave it to customers who bought ASDG hardware). But I had made a promise to myself that someday I would make it available to the whole Amiga community.

I'm really glad I did because that, in and of itself, gave me the personal satisfaction far beyond anything in my history of being involved with computers. It was something, for the first time, that I wrote that thousands of people were using, and they liked it. Every time the RAM disk recovers, I still think it's magic.

The public has responded—we've got 400 people who have sent in their $10. If I had to do it over again, maybe I would have made it a commercial product. However, it certainly wouldn't have earned ASDG the positive notoriety that we have received.

**AC:** *VD0: was released as very low-cost shareware. You've recently developed a commercial product, FACC, a floppy disk caching program. This product is non-copy-protected, is being sold at a low cost with an excellent price-value ratio, and has a policy of free, lifetime updates. In spite of this approach, FACC has been widely pirated. What's your response to what is becoming a major problem in the Amiga community?*

**PK:** We are taking a novel approach with the piracy issue. Some people say pirates should be sued or put in jail. Some pirates are just "collectors," and there is nothing you can do about it. However, for most people, there is a legitimate need to determine whether or not you're going to make an informed purchase—whether or not you're going to get quality. More consumer activism is what I think is the answer to piracy. I also think that if we don't do something about piracy on the Amiga, because of the enormous potential of the machine and the expandability of the machine, the software industry will cease to exist.

**AC:** *I'm not sure I'm following you. Could you elaborate?*

**PK:** Software piracy has always been a part of the industry. In the old days, a modem ran at 300 baud and a disk held 200 or 300K. So piracy was a self-limiting thing because you just couldn't do it fast enough.

The Amiga is one of the most enormously powerful machines of any sort. You can put a 9600 baud modem on it, you can put 8 megabytes of memory on it, and you can add 500 megabytes of hard disk.

Now there is a central chain of pirate bulletin boards that runs in an organized fashion, feeding each other at 9600 baud. They, in turn, feed statewide boards who, in turn, feed smaller city and town boards. There is a whole illegal and illicit distribution network. The self-limiting factors of the old days are gone.

In the early days of the Amiga, the people who bought the machine weren't, by their very nature, the type of people who would pirate. They were the most bizarre collection of people ever imagined buying Amigas: video junkies, computer junkies, and nobody in between. Now, with the 500, the general public—those people who had 450 disks for the 64 and hadn't paid for any of them—are now coming to the Amiga. The future of software production for the Amiga is in very serious danger.

**AC:** *What do you see as the answer to the piracy problem?*

**PK:** Recently, on a public conference on People Link, I proposed a contract. For the software developer and distributor, there would be four responsibilities and one right, while the buyer would have four rights and one responsibility.

The only responsibility that the buyer has is to pay for the product, while the only right the developer/distributor has is to be paid for his efforts. Apart from that, the providers of the software are entirely beholden to the buyer. The buyer has a right to get his money's worth, a right to receive a quality product, a right to informative and accurate information before he makes a purchase, and a right to service and support on the product he has bought.

On the converse, the producer has the obligation and responsibility to fulfill the buyer's rights. However, if you take away the one right that the producer has, then it's like sawing off all four legs of a chair—the entire industry falls down.

And who is hurt in the end? Certainly the small producers will be devastated. The big producers will also be hurt—but there you're talking about a less well defined corporate entity, but it still affects individuals, even within a big company. But certainly the consumer will feel the impact if there's nothing good out there. This will happen if there's no incentive to produce good software.

And here's another interesting observation. Piracy favors the producers of junk, because the people who produce junk and sell it didn't put a lot of effort into that project. So any money they get back is all gravy. For the people who produce quality, it doesn't come free, because you've got to spend time to produce quality. They will not recoup their investment, and they won't do it again because it isn't profitable. So, in the end, it isn't profitable to produce quality; it's only profitable to produce junk.

A lot of people ask, "What should we do about pirates?" My opinion is that we should increase people's awareness, let them know the consequences of what they are doing. This also means not only increasing the awareness of the consumer so that they don't pirate but also increasing their awareness so they demand service. They are entitled to it because they paid their money. I don't think manufacturers should be in business strictly to make money if it is done at the expense of providing service. If you can't do both together, you shouldn't be in business.

**AC:** *On the subject of service, there is a perception among Amiga consumers that ASDG offers both service and quality in their hardware products. At the same time, however, the comment is often made that your prices are high in comparison to similar products. How would you respond to this complaint?*

**PK:** There are some other companies that do things that, quite frankly, we would never have thought of. We just don't know how to run a business that way. Our stuff always seems to come out more expensive than everybody else's. Our training is to produce quality stuff, as good quality stuff as we can produce.

Our background before coming to the Amiga was in the design of high-performance Unix workstations. When somebody spends $400,000 on a purchase, they don't want to hear any excuses—for that amount, they expect the equipment to provide the services they require and for which they paid.

In the Amiga marketplace, we believe the consumer is unbelievably undemanding. They are being abused heavily, and nobody bats an eyelash. I see a lot of companies in the Amiga marketplace who don't share that opinion. I think the Amiga consumer is being far too passive. I can't tell you how many times I've heard a consumer say, "I needed product x, so on July 5th and I ordered it. Today is July 9th and I got it and now I'm really happy." But he ordered it July 5th, 1986 and he got it July 9th, 1987! It took him a year to get it and he's really happy!? Or "I'm happy now, but I had to return to product seven times." So because of the shipping costs, he has had to pay double what the product was worth. I don't buy that.

The Amiga represents to us the first opportunity where we could manufacture our own design. So we got into it wholeheartedly. ASDG was started on my entire life savings. We are now only doing Amiga products. The things that we were doing before, we're not doing any more. I think a lot of people will say about ASDG that the company is run the way *they* would run the company. That's at least what I would hope they would say, and they are exactly right.

The way we do it is quality first. If it's not high quality, we won't do it. If it's not high quality, we won't sell it. And if it's not high quality, we won't put our names on it. And that's it. That's first and last and only.

That has some secondary effects, in that our stuff has to be a little more expensive. We face very fierce competition from people who provide inferior products and even dangerous products. By and large in the Amiga hardware market, people out there are producing good stuff. Some are cutting a few corners. Some are cutting no corners. We cut no corners. We don't know how; that's not what our training is. We don't do it. But there are some companies that cut corners that shouldn't be cut.

**AC:** *Could you expand on that comment?*

**PK:** They produce stuff which is dangerous and that will blow up machines, especially on the 500. You will see 500s starting to blow up or smoke rising from them. I mean that seriously. The 1000 is one of the most forgiving machines you can imagine. It has an 8 amp power supply, and the Amiga consumes 2.5 to 3 amps. So you have 5 amps of slack. The 500 consumes 4.5 amps, and the power supply is 4.5 amps. There are some companies that are selling internal boards that consume 2.5 amps. So you're talking about putting 7 amps on a 4.5 amp computer. And then they say they guarantee they have the best stuff. It's false. It's not fair to the consumer, and it's not fair to us. I've had this conversation with Byte by Byte and CSA and everyone else whose products the public says are too expensive. We all feel the same way—if people knew what they were getting, then they would pay the cost happily. And if they knew what they weren't getting, then they wouldn't buy the other products.

**AC:** *With the introduction of the 2000, Commodore also introduced a new form factor for Zorro cards. In addition, the 500 saw the expansion port moved from the right to the left side of the machine. As a hardware manufacturer, how did you view these changes?*

**PK:** For ASDG, as soon as the 2000 was announced, it put us in a really bad financial position. We had just got our products for the 1000 out, and now we suddenly had to invest all the working capital we had into re-laying out and reproducing products we already had developed. If the 2000 had not come out, ASDG would be a lot bigger company than it is now. But we've managed to survive, and we're growing. Nonetheless, the day we heard about the 2000 we recognized that it was the future and that it was a good idea. With the 1000, the single largest mistake was not having internal expansion. The 2000 fixed that very well. If it had been the old style Zorro boards, it would have been better.

The 2000, by the way, put some companies out of business. Some companies couldn't make the cut or the crossover. Certainly the real early developers like MicroForge and Tecmar— they're history. I would really like to know why. What did they do or didn't do that set them up for the termination of their enterprises?

As for impact of having three different machines—well, for the software manufacturers, it's been wonderful. For us, it was at one time a problem, but it's a problem that won't happen again. At least it damn well better not happen again. In the design of the 2000, anything that is needed in the future can be incorporated in its design.

**AC:** *What is your general view of the Amiga's future?*

**PK:** The 500 and the 2000 are very important steps. The 500 represents both good and bad things.

It represents great things in that it is the most potent package of computing horsepower for its cost ever made. Bar none. Everybody knows that. It represents bad things because as I was talking about before, it is so powerful and so cheap that the Golden Era of just video junkies, art junkies, and computer junkies is ending. Everybody will be buying it, and in some sense, it will be "spoiling the neighborhood." But we couldn't go anywhere if it was just the "junkies."

The 2000 now and for a long time to come is a very important factor in the long term success of the Amiga. If the 2000 had not come out, the 500 would have been an enormously successful home machine, but all it would have been was a game machine. It never would have been a business computer or a vertical market or a scientific academic computer. With the 2000, it has the possibility to be all those things. And on the other end, the 500 has the consumer market sewn up. So we've been behind the 2000 since the day it came out.

**AC:** *Do you see the point in developing a machine beyond the 2000?*

**PK:** Definitely. As a consumer machine, when the price of computing horsepower drops, everybody will have a 68020 at home. It's really a question, though, of where Commodore wants to go with it. Because you have to remember what their corporate history is. Their background is in the production of mass market machines.

Do they want to be in competition with people like DEC and Sun? Because that's where they are heading. Or will they play to their enormous strength, which is in the mass market? So that's a question that has yet to be answered. Whether a 68020 machine comes out from Commodore is a question that has to be answered by the very highest levels of Commodore management—by Irving Gould and God himself.

**AC:** *Given ASDG's previous involvement in designing high end machines, how do you think the Amiga would do in that market?*

**PK:** I don't believe Commodore should go into high end workstations. The competition there is already too fierce. It may come as a surprise to many people, but you can get a Sun workstation for just a little bit more than an Amiga 2000.

That doesn't mean that the 500 is the future of Commodore. The 2000 is there, too. The 2000 is still at a price point were it represents a package that is enormously powerful for the money that you pay. But if they stick their necks above the clouds represented by the $5000 barrier, I don't think they will be equipped to compete against the people who are there. Everybody else is already in the $5000 and above market, but nobody else can follow Commodore down to the $1000 and below market. The only other people are Atari—but who cares. I wouldn't consider the ST as a serious computer.

I do consider the Amiga a damn serious computer. I wouldn't have gotten one otherwise. I wouldn't have gotten the Amiga if I couldn't do the same things I was doing on my VAX. And now I like my Amiga better.

**AC:** *You have some new products coming out in the near future. Could you tell us about them?*

**PK:** We are producing the 2001 expansion box because we felt we had a responsibility. We don't expect to sell very many of them. If we do, we'll be very happy. But we promised our customers an upgrade, so we're going through the development and production expenses because of our promise to our Amiga 1000 owners to do something to keep them functionally current. And people ask me every time, "What do I get for $799? Gee, it has no clock, no memory! It's just a box!" Well, unless you're building billions and billions of boxes, you have no idea how expensive they are. Mechanical things are phenomenally expensive because it's all manual labor.

The SDP is, in my humble opinion, the single most important thing that has happened to the Amiga—in at least the last ten minutes! Seriously, it is really an important product for the Amiga.

What do you get on the average hard disk? 50,000 bytes per second peak or even maximum? With the SDP, we're talking upwards of 400,000 bytes per second. This controller, in the multibus and the VME market where we came from, is a $3000 controller. No punches pulled, and you can check out any magazine.

In the Amiga market, it will be priced in the neighborhood of $700 to $800. We will be able to handle 56 SCSI devices. If you were doing free format I/O at more than a megabyte per second average, not peak, we could use the Amiga as a real time digital video tape editing system.

You could use Amigas as effective file servers for much larger machines. The possibilities are endless. Previously when you put a 68020 on a 2000, you had a machine that computed as fast as a VAX, but its I/O was bull-puckey because you couldn't squeeze the bits in and out fast enough. The SDP will kick the pants off a VAX's I/O because they suffer the same sort of problems without a special controller like the SDP. The Unix environment imposes special restrictions on how fast you can do things because of the way the software is structured.

So now with a 68020 and the SDP, you can have the equivalent of a VAX 8600 or a Sun 3. You have a low cost package that blows the doors off a $400,000 computer. I don't mean to be vain or conceited or have an enlarged idea of what ASDG can do, but I very seriously doubt that there are more than one or two companies in the Amiga marketplace today who could have done an SDP.

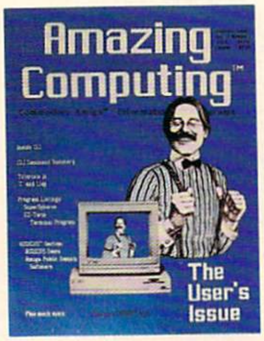
**AC:** *So what does the future hold for ASDG?*

**PK:** We are going to continue to play to our strengths, which are in high end design. When we first came out with memory boards—well memory is memory—it runs at 7 Mhz. with no wait states. Memory is like white bread—it's a commodity. We tried to talk to people about quality, but that's really a secondary issue when it comes to memory. It's all based on price. In the future, we plan to do the things that only we or a very few other companies can do. If you think about what that might involve, you can probably predict what we're going to work on next.

**AC:** *Well, I'm sure that statement will get some imaginations going. I'm equally sure we can look forward to some interesting developments from both you and ASDG. It certainly has been a pleasure talking to you.*

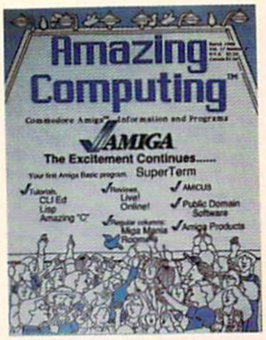
•AC•

# *Expanding Reference*



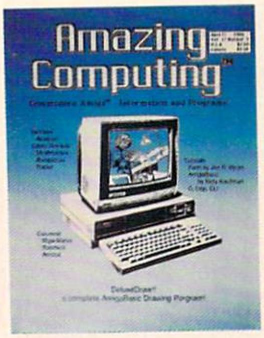
### Volume 1 Number 1 Premiere 1986

**Super Spheres** By Kelly Kauffman An ABasic Graphics prog.
**Data Virus** By J Foust A disease may attack your Amiga!
**EZ-Term** by Kelly Kauffman An ABasic Terminal program
**Miga Mania** by P. Kivolowitz Programming fixes & mouse care
**Inside CLI** by G. Musser a guided insight into the AmigaDos™
**CLI Summary** by G. Musser Jr. A list of CLI commands
**AmigaForum** by B. Lubkin Visit Compuserve's Amiga SIG
**Commodore Amiga Development Program** by D. Hicks
**Amiga Products** A listing of present and expected products


Amazing Computing
Commodore Amiga Information and Programs
AMIGA
The Excitement Continues......
SuperTerm

### Volume 1 Number 2 March 1986

**Electronic Arts Comes Through** A review of software from EA
**Inside CLI: part two** G. Musser Investigates CLI & ED
**A Summary of ED Commands**
**Live!** by Rich Miner A review of the Beta version of Live!
**Online and the CTS Fabite 2424 ADH Modem** by J. Foust
**Superterm V 1.0** By K. Kauffman A term. prog. in Amiga Basic
**A Workbench "More" Program** by Rick Wirch
**Amiga BBS numbers**



### Volume 1 Number 3 April 1986

**Analyze!** a review by Ernest Viverios
**Reviews of Racter, Barataccas and Mindshadow**
**Forth!** The first of our on-going tutorial
**Deluxe Draw!!** by R. Wirch An Amiga Basic art program
**Amiga Basic**, A beginners tutorial
**Inside CLI: part 3** by George Musser George gives us PIPE

Amazing Computing
Commodore Amiga Information and Programs
SCRIMPER!
Screen Image Printer in C!
Our First
Amazing Hardware Project
Connect your own IBM 5 1/4 Drive

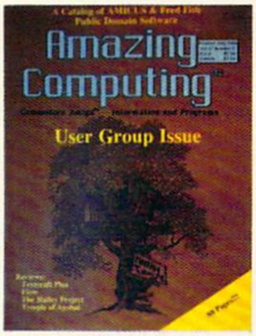
### Volume 1 Number 4 May 1986

**SkyFox and Articfox Reviewed**
**Build your own 5 1/4 Drive Connector** By Ernest Viveiros
**Amiga Basic Tips** by Rich Wirch
**Scrimper Part One** by P. Kivolowitz prog to print Amiga screen
**Microsoft CD ROM Conference** by Jim O'Keane
**Amiga BBS Numbers**

Amazing Computing
Commodore Amiga Information and Programs
COMDEX Spring
Sidecar A First Look
An Interview with R. J. Mical

### Volume 1 Number 5 1986

**The HSI to RGB Conversion Tool**
by S. Pietrowicz Color manipulation in BASIC
**AmigaNotes** by Rick Rae The first of the Amiga music columns
**Sidecar A First Look** by John Foust A first "under the hood"
**John Foust Talks with R. J. Mical at COMDEX™**
**How does Sidecar affect the Transformer**
an interview with Douglas Wyman of Simile
**The Commodore Layoffs** by J. Foust A look Commodore "cuts"
**Scrimper Part Two** by Perry Kivolowitz
**Marauder reviewed** by Rick Wirch
**Building Tools** by Daniel Kary



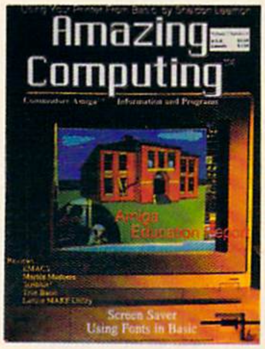
### Volume 1 Number 6 1986

**Temple of Apshai Triology** reviewd by Stephen Pietrowicz
**The Halley Project: A Mission in our Solar System**
reviewed by Stephen Pietrowicz
**Flow:** reviewed by Erv Bobo
**Textcraft Plus a First Look** by Joe Lowery
**How to start your own Amiga User Group** by William Simpson
**Amiga User Groups**
**Mailing List** by Kelly Kauffman a basic mail list program
**Pointer Image Editor** by Stephen Pietrowicz
**Scrimper: part three** by Perry Kivolowitz
**Fun With the Amiga Disk Controller** by Thom Sterling
**Optimize Your AmigaBasic Programs for Speed** by Pietrowicz

Amazing Computing
Commodore Amiga Information and Programs
TWO Complete 3D Graphics Programs
Graphic Tools
AEGIS

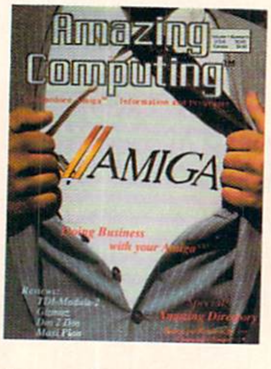
### Volume 1 Number 7 1986

**Aegis Draw: CAD comes to the Amiga** by Kelly Adams
**Try 3D** by Jim Meadows an introduction to 3D graphics
**Aegis Images/ Animator:** a review by Erv Bobo
**Deluxe Video Construction Set** reviewed by Joe Lowery
**Window requesters in Amiga Basic** by Steve Michel
**ROT** by Colin French a 3D graphics editor
**"I C What I Think"** Ron Peterson with a few C graphic progs
**Your Menu Sir!** by B Catey program Amiga Basic menues
**IFF Brush to AmigaBasic 'BOB'** Basic editor by M Swinger
**Linking C Programs with Assembler Routines on the Amiga**
by Gerald Hull



### Volume 1 Number 8 1986

**The University Amiga** By G.Gamble
Amiga's inroads at Washington State University
**MicroEd** a look at a one man army for the Amiga
**MicroEd, The Lewis and Clark Expedition** reviewed Frizelle
**Scribble Version 2.0** a review
**Computers in the Classroom** by Robert Frizelle
**Two for Study** by Frizelle Discovery & TheTalking Coloring Book
**True Basic** reviewed by Brad Grier
**Using your printer with the Amiga**
**Marble Madness** reviewed by Stephen Pietrowicz
**Using Fonts from AmigaBasic** by Tim Jones
**Screen SaVer** by P. Kivolowitz A monitor protection prog. in C
**Lattice MAKE Utility** reviewed by Scott P. Evernden
**A Tale of Three EMACS** by Steve Poling
**.bmap File Reader in Amiga Basic** by T Jones

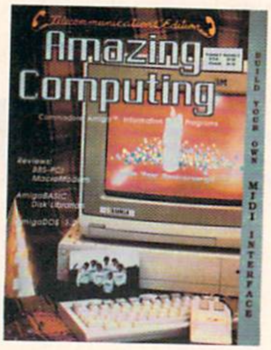


### Volume 1 Number 9 1986

**Instant Music** Reviewed by Steve Pietrowicz
**Mindwalker** Reviewed by Richard Knepper
**The Alegra Memory Board** Reviewed by Rich Wirch
**TxEd** Reviewed by Jan and Cliff Kent
**Amazing Directory** A guide to the sources and resources
**Amiga Developers** A listing of Suppliers and Developers
**Public Domain Catalog** A listing of Amicus and Fred Fish PDS
**Dos 2 Dos** review R. Knepper
Transfer files from PC/MS-DOS and AmigaBasic
**MaxiPlan** review by Richard Knepper The Amiga Spreadsheet
**Gizmoz** by reviewed by Peter Wayner Amiga extras!
**The Loan Information Program** by Brian Catley
basic prog. to for your financial options
**Starting Your Own Amiga Related Business** by W. Simpson
**Keep Track of Your Business Usage for Taxes** by J. Kummer
**The Absoft Amiga Fortran Compiler** reviewed by R A. Reale
**Using Fonts from AmigaBasic, Part Two** by Tim Jones
**68000 Macros on the Amiga** by G. Hull Advance your ability.
**TDI Modla-2 Amiga Compiler** review by S Faiwisze

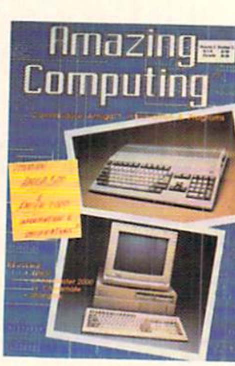
### Volume 2 Number 1 1987

**What Digi-View is... Or, What Genlock Should Be!** by J. Foust
**AmigaBasic Default Colors** by Bryan Catley
**AmigaBasic Titles** by Bryan Catley
**A Public Domain Modula-2 System** reviewed by Warren Block
**One Drive Compile** by Douglas Lovell
Using Lattice C with a single drive system
**A Megabyte Without Megabucks** by Chris Irving
An Internal Megabyte upgrade
**Digi-View** reviewed by Ed Jakober
**Defender of the Crown** reviewed by Keith Conforti
**Leader Board** reviewed by Chuck Raudonis
**Roundhill Computer System's PANEL** reviewed by Ray Lance
**Digi-Paint....by New Tek** previewed by John Foust
**Deluxe Paint II ....from Electronic Arts** previewed by J. Foust



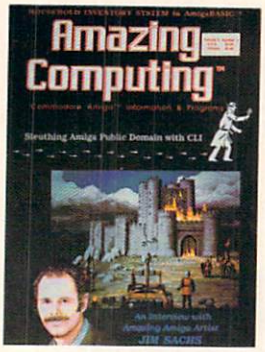
### Volume 2 Number 2 1987

**The Modem** by Josph L. Rothman efforts of a BBS Sysop
**MacroModem** reviewed by Stephen R. Pietrowicz
**GEMINI or "It takes two to Tango"** by Jim Meadows
Gaming between machines
**BBS-PC!** reviewed by Stephen R. Pietrowicz
**The Trouble with Xmodem** by Joseph L. Rothman
**The ACO Project....Graphic Teleconferencing on the Amiga**
by S. R. Pietrowicz
**Flight Simulator II...A Cros Country Tutorial** by John Rafferty
**A Disk Librarian in AmigaBASIC** by John Kennan
**Creating and Using Amiga Workbench Icons** by C. Hansel
**AmigaDOS version 1.2** by Clifford Kent
**The Amazing MIDI Interface build your own** by Richard Rae
**AmigaDOS Operating System Calls and**
**Disk File Management** by D. Haynie
**Working with the Workbench** by Louis A. Mamakos Prog in C



### Volume 2 Number 3

**The Amiga 2000™** by J Foust
A First look at the new, high end Amiga™
**The Amiga 500™** by John Foust
A look at the new, low priced Amiga
**An Analysis of the New Amiga PCs** by J. Foust
Speculation on the New Amigas
**Gemini Part II** by Jim Meadows
The concluding article on two-player games
**Subscripts and Superscripts in AmigaBASIC** by Ivan C. Smith
**The Winter Consumer Electronics Show** by John Foust
**AmigaTrix** by W. Block Amiga™ shortcuts
**Intuition Gadgets** by Harriet Maybeck Tolly
A journey through gadget-land, using C
**Shanghai reviewed** by Keith M. Conforti
**Chessmaster 2000 & Chessmate** reviewedby Edwin V. Apel, Jr.
**Zing! from Meridian Software** reviewed by Ed Bercovitz
**Forth!** by Jon Bryan Get stereo sound into your Forth programs.
**Assembly Language on the Amiga™** by Chris Martin
**Roomers** by theBandito Genlocks are finally shipping, & MORE!!!
**AmigaNotes** by R. Rae Hum Busters... "No stereo? Y not?...
**The AMICUS Network** by J. Foust
CES, user group issues and Amiga Expo"

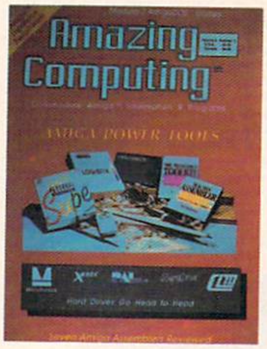

Household Inventory System in AmigaBASIC
Amazing Computing
Sleuthing Amiga Public Domain with CLI
An Interview with Amazing Amiga Artist JIM SACHS

### Volume 2 Number 4 1987

**Amazing Interviews Jim Sachs** by S. Hull Amiga Artist
**The Mouse That Got Restored** by Jerry Hull and Bob Rhode
**Sluething Public Domain Disks with CLI** by John Foust
**Highlights from the San Francisco Commodore Show**
by S Hull
**Speaker Sessions: San Francisco Commodore Show** H Tolly
**The Household Inventory System in AmigaBASIC™**
by B Catley
**Secrets of Screen Dumps** by Natkun Okun
**Using Function Keys with MicroEmacs** by Greg Douglas
**Amigatrix II** by Warren Block More Amiga shortcuts
**Basic Gadgets** by Brian Catley Create gadget functions
**Gridiron** reviewed by K. Conforti Real football for the Amiga
**Star Fleet I Version 2.1** reviewed by J. Tracy Amigain Space
**The TIC** reviewed by J. Foust Battery powered Clock Calendar
**Metascope** review by H. Tolly An easy-to-use debugger

### Volume 2 Number 5 1987

**The Perfect Sound Digitizer** review by R. Battle
**The Future Sound Digitizer** by W. Block Applied Vision's SD
**Forth!** by J. Bryancomparing JForth amd Multi-Forth.
**Basic Input** by B. Catley AmigaBASIC input routine for use in
all your programs.
**Writing a SoundScape Module** in C by T. Fay Programming
with MIDI, Amiga and SoundScape by SoundScape author.
**Programming in 68000 Assembly Language** by C. Martin
Continuing with Counters & Addressing Modes.
**Using FutureSound with AmigaBASIC** by J. Meadows
AmigaBASIC Programming utility with real, digitized STEREO
**AmigaNotes** by R. Rae A review of Mimetics
SoundScape Sound Sampler.
**More AmigaNotes** by R. Rae
A further review of Sunrize's Perfect Sound .
**Waveform Workshop in AmigaBASIC** by J. Shields edit & save
waveform for use in other AmigaBASIC programs.
**The Mimetics Pro MIDI Studio** by Sullivan, Jeffery
A review of Mimetics' music editor/player.
**Intuition Gadgets Part II** by H. MaybeckTolly Boolean gadgets
provide the user with an on/off user interface.



### Volume 2 Number 6 1987

**Forth!** by J. Bryan Access resources in the ROM Kernal.
**The Amazing Computing Hard Disk Review** by J. Foust & S. Leemon In-depth looks at the C Ltd. Hard Drive, Microbotics' MAS-Drive20, Byte by Byte's PAL Jr., Supra's 4x4 Hard Drive and Xebec's 9720H Hard Drive. Also, a look at disk driver software currently under devlopment.

**Modula-2 AmigaDOS™ Utilties** by S. Faiwiszewsk A
Calls to AmigaDOS and the ROM kernal.
**Amiga Expansion Peripheral** by J. Foust
Explanation of Amiga expansion peripherals.
**Amiga Technical Support** by J. Foust
How and where to get Amiga tech support.
**Goodbye Los Gatos** by J. Foust Closing Los Gatos.
**The Amicus Network** by J. Foust West Coast Computer Faire.
**Metacomco Shell and Toolkit** by J. Foust A review
**The Magic Sac** by J. Foust Run Mac programs on your Amiga.
**What You Should Know Before Choosing an Amiga 1000**
**Expansion Device** by S. Grant
**7 Assemblers for the Amiga** by G. Hull Choose your assembler
**High Level Shakeup Replaces Top Management at**
**Commodore** by S. Hull
**Peter J. Baczor"**by S. Hull Manager at CBM gives an inside look
**Logistix** A review by Richard Knepper
**Organize!** by A review Richard Knepper database.
**68000 Assembly Language Programming on the Amiga**
by Chris Martin
**Superbase Personal Relational Database** by Ray McCabe
**AmigaNotes** by Rae, Richard A look at FutureSound
**Commodore Shows the Amiga 2000 and 500 at the Boston**
**Computer Society** by H Maybeck Tolly

Amazing Computing
Commodore Amiga™ Information & Programs
AMAZING VIDEOS
An inside look of the CSA Turbo-Amiga!

### Volume 2, Number 7 1987

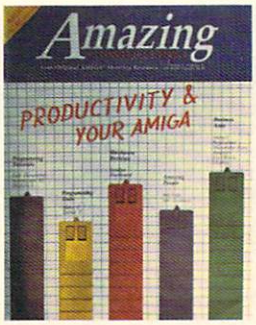
**New Breed of Video Products** by John Foust..
**Very Vivid!** by Tim Grantham...
**Video and Your Amiga** by Oran Sands III
**Amigas & Weather Forecasting** by Brenden Larson
**A-Squared and the Live! Video Digitizer** by John Foust.
**Aegis Animator Scripts and Cel Animation** by John Foust
**Quality Video from a Quality Computer** by Oran Sands III.
**Is IFF Really a Standard?** by John Foust..
**Amazing Stories and the Amiga™** by John Foust.
**All about Printer Drivers** by Richard Bielak
**Intuition Gadgets** by Harriet Maybeck Tolley.
**Deluxe Video 1.2** by Bob Eller
**Pro Video CG1** by Oran Sands III.
**Digi-View 2.0 Digitizer/Software** by Jennifer M. Janik
**Prism HAM Editor from Impulse** by Jennifer M. Janik
**Easyl drawing tablet** by John Foust..
**CSA's Turbo-Amiga Tower** by Alfred Aburto
**68000 Assembly Language** by Chris Martin.

### Volume 2, Number 8 1987

This month Amazing Computing™ focuses on entertainment packages for the Amiga. Amazing game reviews...

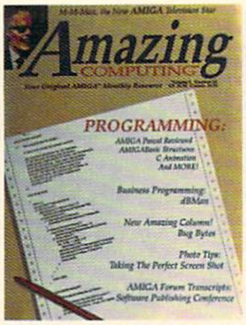
SDI, Earl Weaver Baseball, Portal, The Surgeon, Little Computer People, Sinbad, StarGlider, King's Quest I, II and III, Faery Tale Adventure, Ultima III, Facets of Adventure, Video Vegas and Bard's Tale.

**Plus Amazing monthly columns...** Amiga Notes, Roomers, Modula-2, 68000 Assembly Language and The Amicus Network.
**Disk-2-Disk** by Matthew Leeds
**The ColorFonts** Standard by John Foust
**Skinny C Programs** by Robert Riemersma, Jr.
**Hidden Messages** In Your Amiga™ by John Foust
**The Consumer Electronics Show and Comdex** by J Foust



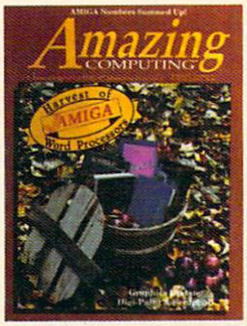
### Volume 2 Number 9 1987

**Analyze 2.0** reviewed by Kim Schaffer
**Impact Business Graphics** review by Chuck Raudonis
**Microfiche Filer** review by Harv Laser
**Pagesetter** review by Rick Wirch
**Gizmoz Productivity Set 2.0** review by Bob Eller
**Kickwork** review by Harv Laser
**Diga Telecommunications Package** review by Steve Hull
**Mouse Time and Timesaver** review by John Foust
**Insider Memory Expansion** review by James O'Keane
**Microbotics Starboard-2** review by S. Faiwiszewski
**Leather Goddesss of Phobos**
reviewed by Harriet Maybeck-Tolly
**Lattice C Compiler Version 3.10** reviewed by Gary Sarff
**Manx 3.4a Update** reviewed by John Foust
**AC-BASIC** reviewed by Sheldon Leemon
**AC-BASIC Compiller** an alternative comparison by B Catley
**Modula-2 Programming** S Faiwiszewski
Raw Console Device Events
**Directory Listings Under AmigaDOS** by Dave Haynie
**AmigaBASIC Patterns** by Brian Catley
**Programming with Soundscape**
Todor Fay manipulate's samples
**Bill Volk, Vice-President Aegis Development,**
interviewed by Steve Hull
**Jim Goodnow, Devekoper of Manx 'C'**
interview by Harriet M Tolly


Amazing
COMPUTING
PROGRAMMING:
Business Programming: dBMan
New Amazing Column! Bug Bytes
Photo Tips: Taking The Perfect Screen Shot
AMIGA Forum Transcripts: Software Publishing Conference

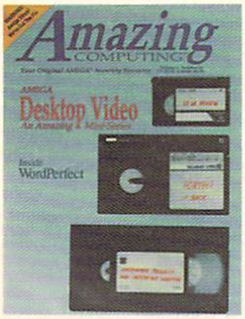
### Volume 2 Number 10 1987

**Max Headroom and the Amiga** by John Foust
**Taking the Perfect Screen Shot** by Keith Conforti
**Amiga Artist: Brian Williams** by John Foust
**Amiga Forum on Compuserve™... Software Publishing Conference Transcript** by Richard Rae
**All About Online Conferencing** by Richard Rae
**dBMAN** reviewed by Clifford Kent
**Amiga Pascal** reviewed by Michael McNeil
**AC-BASIC Compiler** reviewed by Bryan Catley
**Bug Bytes** by John Steiner
**Amiga Notes** by Richard Rae
**Roomers** by The Bandito
**68000 Assembly Language** by Chris Martin
**The AMICUS Network** by John Foust
**Amiga Programming:**
**Amiga BASIC Structures** by Steve Michel
**Quick and Dirty Bobs** by Michael Swinger
**Directory Listings Under Amiga-DOS, Part II** by Dave Haynie
**Fast File I/O with Modula-2** by Steve Faiwiszewski
**Window I/O** by Read Predmore



### Volume 2 Number 11 1987

**Word Processors Rundown** by Geoff Gamble
ProWrite, Scribble!, and WordPerfect compared
**LPD Writer Review** by Marion Deland
**VizaWrite Review** by Harv Laser
**Aedit Review** by Warren Block
**WordPerfect Preview** by Harv Laser
**Jez San Interview** by Ed Bercovitz
The author of StarGlider speaks!
**Do-it-yourself Improvements to the Amiga Genlock**
**Digi-Paint Review** by Harv Laser
**Sculpt 3D Review** by Steve Pietrowicz
**Shadowgate Review** by Linda Kaplan
**TeleGames Review** by Michael T. Cabral
**Reason Preview**: a quick look at an
intense grammar examination application
**As I See It** by Eddie Churchill Peeking at WordPerfect,
Gizmoz V2.0 and Zing! Keys
**Bug Bytes** by John Steiner
**AmigaNotes** by R Rae 4 electronic music books
**Modula-2 Programming** by Steve Faiwiszewski
devices, I/O, and the serial port
**Roomers** by The Bandito
**68000 Assembly Language** by Chris Martin
Chris walks through the display routines
**The AMICUS Network** by John Foust
Desktop Publishing & Seybold
**C Animation Part II** by Mike Swinger Animation Objects
**BASIC Text** by Brian Catley Pixel perfect text positioning
**Soundscape Part III** by Todor Fay VU Meter and more
**Fun with Amiga Numbers** by Alan Barnett
**File Browser** by Bryan Catley
Full Feature BASIC File Browsing utility



### Volume 2 Number 12 1987

**The Ultimate Video Accessory** by Larry White
**The Sony Connection** by Stewart Cobb
**15-Puzzle in AmigaBASIC** by Zoltan Szepsi
**Life, Part I: The Beginning** by Gerald Hull
The ultra-complex nine bit solution to the "Game of Life."
**Amiga Virus!** by John Foust
A new Amiga virus has surfaced. Please check your system.
**CLI Arguments in C** by Paul Castonguay
**MIDI Interface Adapter** by Barry Massoni
Amiga 1000-style MIDI interfaces can fit A2000s or 500s
**Modula-2** by Steve Faiwiszewski
First in a series, a command line calculator in Modula-2.
**AmigaNotes** by Richard Rae
The audio changes made in the Amiga 500 and 2000.
**Animation for C Rookies: Part III** by M. Swinger
tackling double-buffering.
**The Big Picture** by Warren Ring
Amiga™ Assembly language programming for the brave!
**Karate Kid Review**by Stephen R. Pietriowicz
**GO! 64 review** by John Foust, James O'Keane, and Rick Wirch
Three C-64 experts investigate a new Amiga 64 emulator.
**A-Talk-Plus Review** by Brendan Larson
"Full-fledged terminal program" & Tektronics capabilities
**Calligrapher Review** by John Foust
**Animator: Apprentice Review** by John Foust
**Playing Dynamic Drums on the Amiga** by David N. Blank
**WordPerfect Review** by Steve Hul
**Insider/Kwikstart Review** by Ernest P. Viveiros Sr
RAM & ROM expansion: Comments and installation tips.
**Bug Bytes** by John Steiner
**Forth!** by Jon Bryan
DumpRPort utility for your Multi-Forth toolbox.
**As I See It** by Eddie Churchill
An offbeat look on Digi-Paint, Portal, and Videoscape 3D.
**The Amicus Network** by John Foust
The Commodore Show and AmiExpo: New York!

(Back Issue listing continued on page 111)

# *Your Resource to the Commodore Amiga™*

The phrase above is not just empty words. The pages of Amazing Computing™ are filled with articles on technical operations and procedures, basic use, and just-plain-fun. The growing library of Amazing Back Issues contains articles from building your own IBM Disk controller, to setting up your own startup sequence. Amazing Computing™ has repeatedly been the first magazine to offer the Amiga users solid, indepth reviews and hands on articles for their machines.

**Amazing Computing™ was the first magazine to document CLI**
**Amazing Computing™ was the first to show Sidecar™ from COMDEX™ in full detail.**
**Amazing Computing™ was the first to document a 5 1/4 drive connector**
**Amazing Computing™ was the first with a 1 Meg Amiga upgrade hardware project!**
**Amazing Computing™ was the first magazine to offer serious programming examples and help.**
**Amazing Computing™ was the first to offer Public Domain Software at reasonable prices.**
**Amazing Computing™ was the first magazine with the user in mind!**

### From the Beginning

Since February 1986, Amazing Computing™ has been providing users with complete information for their Amiga. This store house of programs and information is still available through our back issues. From the Premiere issue to the present, there are insights into the Amiga that any user will find usefull.

**$4.00 each!**

**Our Back Issue price is still $4.00 per issue!** (Foreign orders, please add $1.00 per issue for Postage & Handling. All payments must be made by check or money order in U.S. funds drawn on a U.S. Bank.)

### Limited Supply

Unforunately, nothing lasts forever, and the availabilty of some of our Back Issues is definitely limited. Please complete your Amazing Computing™ library today, while these issues are still available. Please complete the order form in the rear of this issue and mail with check or money order to:

**Back Issues, PiM Publications, Inc., P.O. Box 869, Fall River, MA 02722**
(Please allow 4 to 6 weeks for delivery)

# Modula-2

by Steve Faiwiszewski

## *Toying with the Gameport Device and Simple Hardware Sprites*

In this article, I'll discuss the gameport device and simple (hardware) sprites, then present a program demonstrating both in action.

### The Gameport Device

The gameport device provides access to the Amiga's two gameports. Unit O controls the front gameport connector; unit 1 controls the other. The front connector (connector 1) is usually used by an input device, so I'll only discuss unit 1.

The gameport device responds to the following commands: GPD_SETCTYPE,GPD_ASKCTYPE, GPD_SETTRIGGER, GPD_ASKTRIGGER, and GPD_READEVENT. These commands are executed through the DoIO or SendIO system functions.

#### GPD_SETCTYPE

This command selects the type of controller the gameport will use. There are currently four types of controllers: mouse, absolute joystick, relative joystick, and "no controller." A mouse controller can report events for as many as three buttons and for positive and negative (x,y) movements. A trackball controller works like a mouse. An absolute joystick reports a single event for each change in its current direction. For example, if the joystick is in the center and the user pushes the stick backward, a backward-switch event is generated. A relative joystick is an absolute joystick with "autorepeat"; as long as the stick is not centered, the gameport continues to generate events.

#### GPD_ASKCTYPE

Use this command to inquire about the gameport's controller type.

#### GPD_SETTRIGGER

This command sets the conditions that trigger a gameport event. It must be used before you issue any read requests. With the mouse, an event can be triggered by a move in either the x or y direction by pressing or releasing the buttons (on a timed basis), or a combination of the two. With a joystick, an event can be triggered by button-up or button-down condition, changes in stick position, or a timed basis. The trigger setting information is placed into a GamePortTrigger record which is declared in the Benchmark Modula-2 package as follows:

```
GamePortTrigger = RECORD
                    gptKeys    : GPTKeysSet;
                    gptTimeout : CARDINAL;
                    gptXDelta  : CARDINAL;
                    gptYDelta  : CARDINAL;
                  END;
```

gptKeys can be set to GPTDownKeys, GPTUpKeys, or both. The gptTimeout field specifies how often timed events should be generated. The timeout is expressed as the number of vertical blank units (1/60th of a second) that should occur before a timed event is triggered.

#### GPD_ASKTRIGGER

Use this command to inquire about the conditions set for triggering.

#### GPD_READEVENT

This reads one or more gameport events. The gameport generates an event and writes the information into a data area specified in the read command. (Note that once the device responds to a read request and reports all events that have been queued up, a new read request must be made, or no more events will be generated.) For details on device openings and device I/O, see *Amazing Computing* V2.11 for more information.

**Hardware Sprites**

The Amiga supports eight hardware sprites (simple sprites), as well as many software sprites, known as VSprites (virtual sprites). VSprites are more flexible than simple sprites, and more can be done with them, but they are also more complicated to use. VSprites can be bound to an Intuition screen, but simple sprites cannot. Keep that fact in mind if you decide to use simple sprites because they will appear in front of any screen the user displays, including the Workbench screen. It is a bit strange to use a text editor and see an alien spaceship sitting on top of the text.

A sprite can be up to 16 pixels wide, and as high as the screen display (200 lines). Simple sprites are set up in pairs. When you "steal" a simple sprite, you also steal its color registers. Since both sprites in the pair use the same color registers, you steal the other sprite too—so you might as well ask for it. The sprite color registers are assigned as follows:

| Sprite | Color Registers |
|---|---|
| 0 & 1 | 16–19 |
| 2 & 3 | 20–23 |
| 4 & 5 | 24–27 |
| 6 & 7 | 28–31 |

Registers 16, 20, 24, and 28 always generate the "transparent" color for sprites, regardless of which color they actually contain. (Note that you are not granted exclusive use of the color registers; they can still be used to render into the display area. Exclusive use of a sprite is obtained through the GetSprite call). The procedure is declared as:

```
PROCEDURE GetSprite(VAR sprite : SimpleSprite; pick :
INTEGER): INTEGER;
```

The parameter "sprite" is an instance of the SimpleSprite record to describe the sprite. SimpleSprite is declared as:

```
SimpleSprite = RECORD
                 posctldata  : ADDRESS;
                 height      : CARDINAL;
                 x,y        : CARDINAL;
                 num         : CARDINAL;
               END;
```

"Pick" is the number of the desired sprite. Use –1 if any sprite will do. The call returns the number of the allocated sprite, or –1 if no sprites were allocated.

Once you obtain a sprite, the next thing is to set its shape. To do this, set the height field in the SimpleSprite record, and then call to ChangeSprite, declared as:

```
PROCEDURE ChangeSprite(VAR vp : ViewPort; VAR s:
SimpleSprite, newdata: ADDRESS);
```

"vp" is the viewport structure relative to the sprite. "Newdata" is the address of the sprite data. This consists of two position control CARDINALs, followed by the sprite image data, followed by two CARDINALs reserved for future use. The position control and reserved CARDINALs should be initialized to zero.

Moving hardware sprites is quite simple. Simply call MoveSprite, declared as:

```
PROCEDURE MoveSprite(VAR vp: ViewPort; VAR sprite:
SimpleSprite; x,y: INTEGER);
```

"x,y" are the coordinates to which the sprite should be moved.

Once you are finished using a hardware sprite, you must relinquish control of it. Do this with a call to FreeSprite, declared as:

```
PROCEDURE FreeSprite(num);
```

"num" is the sprite number. If your task does not free it, the sprite will be unavailable until the next reboot.

**The Program**

The accompanying source listings are for a program demonstrating use of a joystick and a hardware sprite. EtchAsketch lets you move a sprite around a screen to draw lines.

The program is made up of four modules: EtchGlobal contains some low level routines used by the other modules. EtchJoystick contains all gameport-related code, and EtchIntuiStuff does all Intuition-related work. The main module contains the code that handles the sprite and draws the lines.

Note the use of the AddTerminatingProc and ExitGracefully procedures. The modula-2 language supports initialization code for each implementation module, but no such feature is available for exit code. These two procedures represent an attempt to create such a feature. This way, each module can designate a procedure which cleans up and releases all resources allocated by the module. Exit from the program is done by calling ExitGracefully, which, in turn, calls all designated terminating procedures.

All memory allocations are remembered by Intuition's AllocRemember routine. This makes it easy to release all allocations simultaneously upon termination of the program.

*[Note: The EtchAsketch described in this article is the author's own program and has no relation to Ohio Art's product of the same name.-Ed.]*

SOURCE BUILDER

Introducing — A CASE Tool for AMIGA

**This powerful new Amiga CASE (Computer Aided Software Engineering) tool is for software designers to automate the creation and maintenance of software from cradle to grave.**

- An effective, affordable, professional CASE tool for automated code generation.
- Easily build user interface and complex data structure routines. Supports reuseable code and object oriented design.
- Logical template substitution for multiple version code generation; i.e. From one common form, generate code for multiple computers, PC, MAC, Amiga & ST.
- Extremely effective in non programming environment; CAD, Wordprocessing, Project Management, and Postscript processing for desk top publishing.
- Allows many ADA tm features to be used in conventional languages such as C. Also extends the object oriented design features of ADA tm.
- Templates and configurations can be easily added and expended by user.
- Includes an user manual as well as a software design tutorial.
- Price $149.95, send check or money order only. CA residents add 6.5% sales tax. This product requires Amiga Workbench 1.2 and Extra disk.

**Software Integration Solutions**
**16496 Bernardo Center Drive**
**San Diego, CA. 92128**
**Tel: 619-451-3094**

## LISTING 1

```
DEFINITION MODULE EtchGlobal;
(*********************************************************)
(*       Some low level routines for EtchAsketch         *)
(*                                                       *)
(*        Written for the Benchmark M2 compiler.         *)
(*                                                       *)
(* Steve Faiwiszewski                    December 1987  *)
(*********************************************************)

FROM Views      IMPORT ViewPortPtr;
FROM Rasters    IMPORT RastPortPtr;
FROM Intuition  IMPORT RememberPtr, ScreenPtr;
FROM SYSTEM     IMPORT ADDRESS;

VAR
   MyScreen    : ScreenPtr;
   MyRPort     : RastPortPtr;
   MyVPort     : ViewPortPtr;
   RKey        : RememberPtr;


PROCEDURE Allocate(VAR ptr : ADDRESS; size : LONGCARD);
(* Allocate memory as MemPublic and clear it.       *)
(* Track is kept of the allocation thru the use of *)
(* Intuition's AllocRemember.                       *)

PROCEDURE ChipAllocate(VAR ptr : ADDRESS; size : LONGCARD);
(* Allocate Chip memory as MemPublic and clear it. *)
(* Track is kept of the allocation thru the use of *)
(* Intuition's AllocRemember.                       *)

PROCEDURE AddTerminationProc(t : PROC);
(* Add procedure t to the list of procedures to be *)
(* called upon graceful exit of the program.        *)

PROCEDURE ExitGracefully;
(* Call all the designated terminating procedures, *)
(* and then halt.                                   *)
END EtchGlobal.
```

## LISTING 2

```
IMPLEMENTATION MODULE EtchGlobal;
(*********************************************************)
(*       Some low level routines for EtchAsketch         *)
(*                                                       *)
(*        Written for the Benchmark M2 compiler.         *)
(*                                                       *)
(* Steve Faiwiszewski                    December 1987  *)
(*********************************************************)

FROM Memory    IMPORT MemReqSet, MemPublic, MemChip,
                      MemClear;
FROM Intuition IMPORT FreeRemember, AllocRemember;
FROM TermInOut IMPORT WriteString, WriteLn;
FROM RunTimeErrors
               IMPORT InstallErrorHandler,
                      RemoveErrorHandler;
FROM SYSTEM    IMPORT ADDRESS;

VAR
    CurrentTermProc : CARDINAL;
    TermProcs       : ARRAY [0..50] OF PROC;

PROCEDURE TheEnd;
BEGIN
    WriteString('Releasing all allocations.'); WriteLn;
    FreeRemember(RKey, TRUE);
    RemoveErrorHandler;
    HALT
END TheEnd;

PROCEDURE Allocate(VAR ptr : ADDRESS; size : LONGCARD);
BEGIN
    ptr := AllocRemember(RKey, size,
                         MemReqSet{MemClear,MemPublic});
END Allocate;

PROCEDURE ChipAllocate(VAR ptr : ADDRESS; size : LONGCARD);
BEGIN
    ptr := AllocRemember(RKey, size,
                    MemReqSet{MemClear,MemPublic,MemChip});
END ChipAllocate;

PROCEDURE AddTerminationProc(t : PROC);
(* Insert procedure t into the array of terminating *)
(* procedures (which are called upon graceful exit. *)
BEGIN
    TermProcs[CurrentTermProc] := t;
    INC(CurrentTermProc)
END AddTerminationProc;

PROCEDURE ExitGracefully;
(* Call all the designated terminating procedures. *)
VAR
    i : CARDINAL;
BEGIN
    FOR i := (CurrentTermProc-1) TO 0 BY -1 DO
        TermProcs[i]
    END
END ExitGracefully;

BEGIN
    RKey := NIL;
    CurrentTermProc := 0;
    AddTerminationProc(TheEnd);
(* The error handler is installed just in case we need it. *)
    InstallErrorHandler;
END EtchGlobal.
```

## LISTING 3

```
DEFINITION MODULE EtchJoystick;
(*********************************************************)
(* Joystick-related routines for the EtchAsketch program *)
(*                                                       *)
(*        Written for the Benchmark M2 compiler.         *)
(*                                                       *)
(* Steve Faiwiszewski                    December 1987  *)
(*********************************************************)
```

```
FROM Tasks IMPORT SignalSet;

CONST
    Forward = -1;
    Backward = 1;
    Left = -1;
    Right = 1;


PROCEDURE OpenJoystick;
(* Open the gameport device and set it up for a joystick *)

PROCEDURE CloseJoystick;
(* Release the gameport resource *)

PROCEDURE PrepareToReadJoystick;
(* Prepare the gameport device for a read. *)

PROCEDURE SetTriggerTime(interval : CARDINAL): BOOLEAN;
(* Set the trigger interval for the joystick. *)
(* Returns true if successful.                *)

PROCEDURE GetJoystickStatus(VAR ButtonDown : BOOLEAN;
                            VAR Xstick, Ystick : INTEGER);
(* Assuming a message came from the joystick          *)
(* (i.e. PrepareToReadJoystick was previously called)  *)
(* examine joystick's direction and button condition   *)

PROCEDURE ListenToJoystick(VAR MySignalSet : SignalSet);
(* Set up the signalset to listen to gameport events *)

END EtchJoystick.
```

## LISTING 4

```
IMPLEMENTATION MODULE EtchJoystick;
(*******************************************************)
(* Joystick-related routines for the EtchAsketch program *)
(*                                                      *)
(*        Written for the Benchmark M2 compiler.        *)
(*                                                      *)
(* Steve Faiwiszewski                   December 1987   *)
(*******************************************************)

FROM EtchGlobal      IMPORT AddTerminationProc,
                            ExitGracefully;
FROM GamePortDevice  IMPORT GPDSetTrigger, GPDSetCType,
                            GPTDownKeys, GPTUpKeys,
                            GPTKeysSet, GamePortTrigger,
                            GPCTNoController, GPCTAbsJoystick,
                            GPDReadEvent;
FROM Ports           IMPORT MsgPortPtr, GetMsg, MessagePtr;
FROM PortsUtil       IMPORT CreatePort, DeletePort;
FROM Tasks           IMPORT SignalSet;
FROM IODevicesUtil   IMPORT CreateStdIO, DeleteStdIO;
FROM IODevices       IMPORT IOStdReqPtr, IOFlagsSet,
                            IOStdReq,
                            DoIO, SendIO, CmdWrite, CmdRead,
                            OpenDevice, CloseDevice;

FROM InputEvents     IMPORT InputEvent, InputEventPtr,
                            IECodeLButton, IECodeUpPrefix;
FROM TermInOut       IMPORT WriteString, WriteLn;
FROM SYSTEM          IMPORT BYTE, ADR, TSIZE, LONGWORD;

CONST
    GamePortName = "gameport.device";

VAR
    JoyMsgPort   : MsgPortPtr;
    JoyIORequest : IOStdReqPtr;
    JoystickData : InputEvent;


PROCEDURE cleanup(line : ARRAY OF CHAR; n : CARDINAL);
BEGIN
    WriteString(line); WriteLn;
    IF n > 2 THEN CloseDevice(JoyIORequest) END;
    IF n > 1 THEN DeleteStdIO(JoyIORequest) END;
    IF n > 0 THEN DeletePort(JoyMsgPort^) END;
END cleanup;

PROCEDURE abort(line : ARRAY OF CHAR; n : CARDINAL);
BEGIN
    cleanup(line,n);
    ExitGracefully
END abort;

PROCEDURE SetTriggerTime(interval : CARDINAL): BOOLEAN;
VAR
    MyTrigger    : GamePortTrigger;
BEGIN
    WITH JoyIORequest^ DO
        ioCommand := GPDSetTrigger;
        ioLength := TSIZE(GamePortTrigger);
        ioData := ADR(MyTrigger);
    END;
    WITH MyTrigger DO
        gptKeys := GPTKeysSet{GPTDownKeys,GPTUpKeys};
        gptTimeout := interval;
        gptXDelta := 1;
        gptYDelta := 1;
    END;
    IF DoIO(JoyIORequest) <> 0D THEN
        RETURN FALSE
    END;
    RETURN JoyIORequest^.ioError = BYTE(0)
END SetTriggerTime;


PROCEDURE SetType(type : BYTE): BOOLEAN;
(* Set the type of gameport controller *)
BEGIN
    WITH JoyIORequest^ DO
        ioCommand := GPDSetCType;
        ioLength := 1;
        ioData := ADR(type);
    END;
    IF DoIO(JoyIORequest) <> 0D THEN
```

**AMIGA DUAL 3½″ DISK DRIVES**

100% Compatable with
Amiga 500, 1000 & 2000 Computers

- Internal Power Supply
- All Metal Chassis
- Horizontal Layout
- Vented Enclosure
- On-Off Switch
- 6 foot 3-prong linecord
- Primary Circuit Breaker Protected
- Color Coordinated to Amiga Computers

**ONLY $395.00**

Single Disk Drive w/pass-thru **$199.00**

**20 Meg Hard Drive (SCSI) with Controller**
**Only $750.00**

***Comp-U-Save***

414 Maple Avenue, Westbury, NY 11590
**In NY State (516) 997-6707**
**Outside NY State (800) 356-9997**

*"Meeting the Needs of People in the Electronic Age"*

```
        RETURN FALSE
    END;
    RETURN JoyIORequest^.ioError = BYTE(0)
END SetType;

PROCEDURE PrepareToReadJoystick;
(* Do an asynchronous I/O (SendIO), telling the *)
(* joystick that we want to read an event.      *)
BEGIN
    WITH JoyIORequest^ DO
        ioCommand := GPDReadEvent;
        ioLength := TSIZE(InputEvent);
        ioData := ADR(JoystickData);
        ioFlags := IOFlagsSet{};
    END;
    SendIO(JoyIORequest)
END PrepareToReadJoystick;

PROCEDURE OpenJoystick;
(* Opening the gameport device requires the creation    *)
(* of a port, allocation of IOStdReq record, and actual *)
(* opening of the device itself.                        *)
BEGIN
    JoyMsgPort := CreatePort(ADR("MyJoyPort"),0);
    IF JoyMsgPort = NIL THEN
        abort("Couldn't create joystick port",0);
    END;
    JoyIORequest := CreateStdIO(JoyMsgPort^);
    IF JoyIORequest = NIL THEN
        abort("Couldn't create joystick io request",1);
    END;
    IF OpenDevice(ADR(GamePortName),1,JoyIORequest,
                  LONGWORD(0)) <> 0D THEN
        abort("Couldn't open joystick device",2)
    END;

(* so far so good.  Now set the controller type   *)
(* to 'joystick'.                                 *)
    IF NOT SetType(BYTE(GPCTAbsJoystick)) THEN
        abort("Couldn't set controller type",3)
    END;
(* We don't want to get timer events as a default *)
    IF NOT SetTriggerTime(0) THEN
        abort("Couldn't set trigger",3)
    END;
    JoyIORequest^.ioMessage.mnReplyPort := JoyMsgPort;
END OpenJoystick;

PROCEDURE GetJoystickStatus(VAR ButtonDown : BOOLEAN;
                            VAR Xstick, Ystick : INTEGER);
BEGIN
    WITH JoystickData DO
        IF ieCode = IECodeLButton THEN
            ButtonDown := TRUE
        ELSIF ieCode = IECodeLButton + IECodeUpPrefix THEN
            ButtonDown := FALSE
        END;
        Ystick := ieY;
        Xstick := ieX;
    END; (* with *)
END GetJoystickStatus;

PROCEDURE ListenToJoystick(VAR MySignalSet : SignalSet);
BEGIN
    INCL(MySignalSet,CARDINAL(JoyMsgPort^.mpSigBit))
END ListenToJoystick;

PROCEDURE CloseJoystick;
(* Set the gameport type to 'no controller' *)
BEGIN
    IF NOT SetType(BYTE(GPCTNoController)) THEN
        WriteString('* Could not reset joystcik port *');
        WriteLn
    END;
    cleanup('Releasing Joystick Resource.',3);
END CloseJoystick;

BEGIN
    AddTerminationProc(CloseJoystick);
END EtchJoystick.
```

•AC•

# HARD DRIVES

PIONEER has the storage solution for your personal computer, business system or graphics workstation. Your requirements are met by our product family. From the 20-megabyte through the high performance 150-megabyte drives, PIONEER spans the storage spectrum to offer the right capacity, performance and price to suit your system needs.

All PIONEER hard drive have an embedded SCSI controller build-in. This makes you compatible with other SCSI systems on the market (no interfaces needed between the computer host controller and your hard drive).

# SCSI HOST CONTROLLER

The SCSI host/controller has been designed with growth and expansion in mind. The hardware and software provided with your SCSI host/controller will allow you to connect, set-up and access information from any SCSI device. The SCSI host/controller allows AmigaDOS to deal with your hard drive at a system level using the same commands that you are now using with the standard Amiga 3.5" floppy drive.

The SCSI host/controller will provide you with the fastest, most reliable and fully featured hard drive system available for the AMIGA 500, 1000 and 2000. The SCSI host/controller comes complete with a expansion bus pass-thru, 18 inch cable, 50-pin SCSI bus and Apple/Mac Pseudo SCSI 25-pin connectors.

# HARD DRIVE CASE

Full-height case comes complete with a 45 watt fan cooled power supply, switchable 110/220 AC voltage (50/60 Hz), room for one full-height or two half-height drives, power-on lED, half-height cover plate and power cord. Dimensions: 7" wide, 5" high and 15.5" deep.

Half-height case comes complete with a 45 watt fan cooled power supply, switchable 110/220 AC voltage (50/60 Hz), room for one half-height drive or tape back-up, convenient carry handle and power cord. Dimensions: 6" wide, 3.25" high and 15.5" deep.

**OUR GUARANTEE TO YOU:**
All our products are 100% guaranteed. We will replace, exchange or refund* any purchase to your satisfaction.
* Refund must be made within 15 days of receipt of product.

**Hard Drive comes complete with drive, SCSI host/controller, your choice of case and a ONE YEAR WARRANTY!**

| | |
|---|---|
| 20 meg | $ 659.95 |
| 40 meg | $ 879.95 |
| 65 meg | $ 949.95 |

## External 3.5" Floppy Drive

PIONEER'S 3.5" floppy drive comes complete with an 18" cable, fully enclosed metal case, a pass-thru for addional drives, single or dual drive, ninety day warranty and consumes about 2.1 watts.

| | |
|---|---|
| Single Drive | $ 159.95 |
| Dual Drive | $ 329.95 |

# PIONEER COMPUTING

*–Blazing a New Frontier for AMIGA COMPUTERS–*

2469 East 7000 South #200
Salt Lake City, UT 84121
(801) 942–1174

# An Interview with Avant Garde Artist, Jean "Moebius" Giraud

*by Edward L. Fadigan*

What do a twisted loop of paper and an airtight garage have in common? They are both creations of Moebius.

The Moebius of the Airtight Garage is French artist Jean Giraud. His nom-de-plume, "Moebius," is only borrowed from the very real German mathematician who created the famous twisted loop of paper known as the "Moebius Strip."

Moebius, the artist, was born in a suburb of Paris on May 8, 1938. He spent a great deal of his childhood with his grandparents, and on their bookshelves, he discovered the works of famous nineteenth century illustrators, such as Gustave Dore. He was fascinated, not only by these classic illustrations, but by comic strips as well, especially *Tarzan, Tintin, Flash Gordon, Spirou, Mandrake,* and *The Phantom.* In fact, he was so impressed by *The Phantom,* he later paid homage to it in his graphic novel, *The Airtight Garage.*

At sixteen, Giraud began artistic studies at the Arts Appliques, a professional school in Paris. Around this time, he discovered the fascinating concepts being explored in science fiction, through the French edition of *Fantasy & Science Fiction.*

Besides having been published in many science fiction magazines, such as the European *Metal Hurlant* and its American counterpart, *Heavy Metal,* Moebius has lent his artistic hand to several science fiction films as well. He worked on *Alien,* designing a number of spacesuits and costumes. His influence can also be seen in *TRON,* the first film to use large amounts of computer animation.

In the past few years, Moebius has to contributed designs to the live-action feature *Masters of the Universe,* and has worked on *Willow,* a live-action fantasy picture to be produced by George Lucas and directed by Ron Howard.

Here is Moebius' work (on page 57), presented with a brand-new tool, the Amiga computer. Comparing this to his more traditional artwork (on page 56), it is easy to see that the Amiga in no way restricts an artist's style. This work is most definitely "Moebius."

***AC:*** *Tell me how you got involved with the Amiga, and computer graphics.*

***MOEBIUS:*** Well, after I bought the Amiga for my son Julien, he showed me all the really nice graphics it could do. It was then that I started to experiment with it. Once in a while I would sit down and play, just drawing computer pictures for the fun of it. Now that I feel more comfortable using it, I'd like to try some animation. Julien just got "VideoScape 3D." I'm really looking forward to using that one, but I'll have to wait until he knows it well enough to show me how to use it.

***AC:*** *Did you use any other drawing tools besides the mouse?*

***MOEBIUS:*** No, all of my Amiga drawings so far have been created using only the mouse. We're thinking of getting a drawing tablet soon, though. I think that'll make it a little easier for me.

***AC:*** *Do you find the mouse awkward after years of using traditional tools?*

***MOEBIUS:*** It was a little difficult at first, but after many hours, and a few drawings, I started to get the hang of it. Besides, I like the "rough" look of some of my first computer pictures.

***AC:*** *One of the first was "Black Chair." Tell me about that one.*

***MOEBIUS:*** I did that picture as a kind of technical training. The quality of the finished piece was not my first concern. It was done mostly to get the feel of a new tool. I don't think it came out too bad, though.

***AC:*** *Did you sketch it out and plan it beforehand?*

***MOEBIUS:*** Not at all. I improvised directly on the screen with the paint program. But you're right; it's a good idea to do the initial sketches on paper, and, in the future, I'll be doing more of that.

***AC:*** *What do you think is the future for computer graphics?*

***MOEBIUS:*** I have no doubt it will grow even faster in the future than it is growing today. But I don't think I'll be very involved with it. I still prefer my paper and pencil.

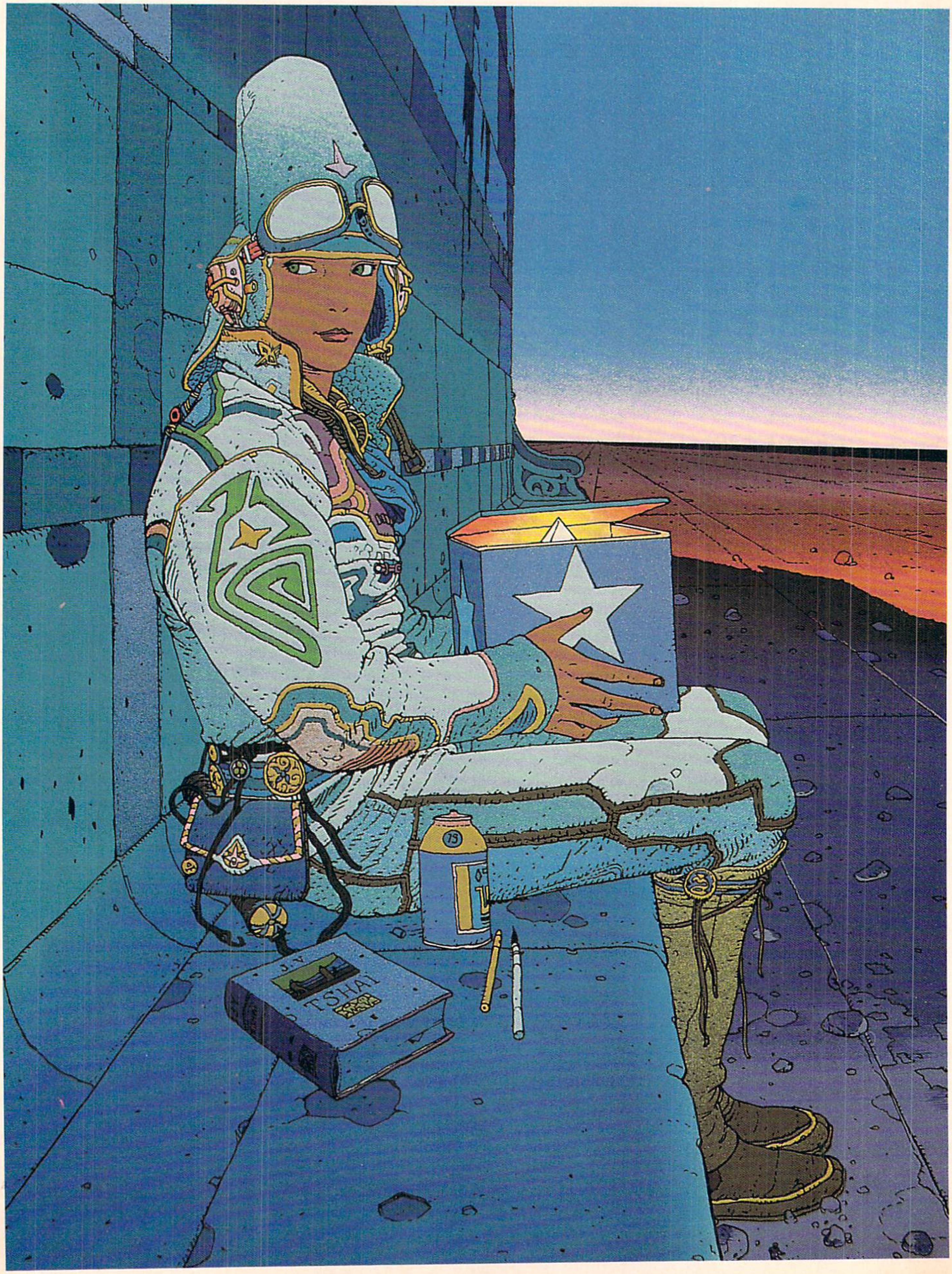
***AC:*** *Do you think computer graphics can be an art in itself, or is it merely an instrument for use in more tangible mediums?*

***MOEBIUS:*** No technique is an art in itself. There are only tools for the creator.

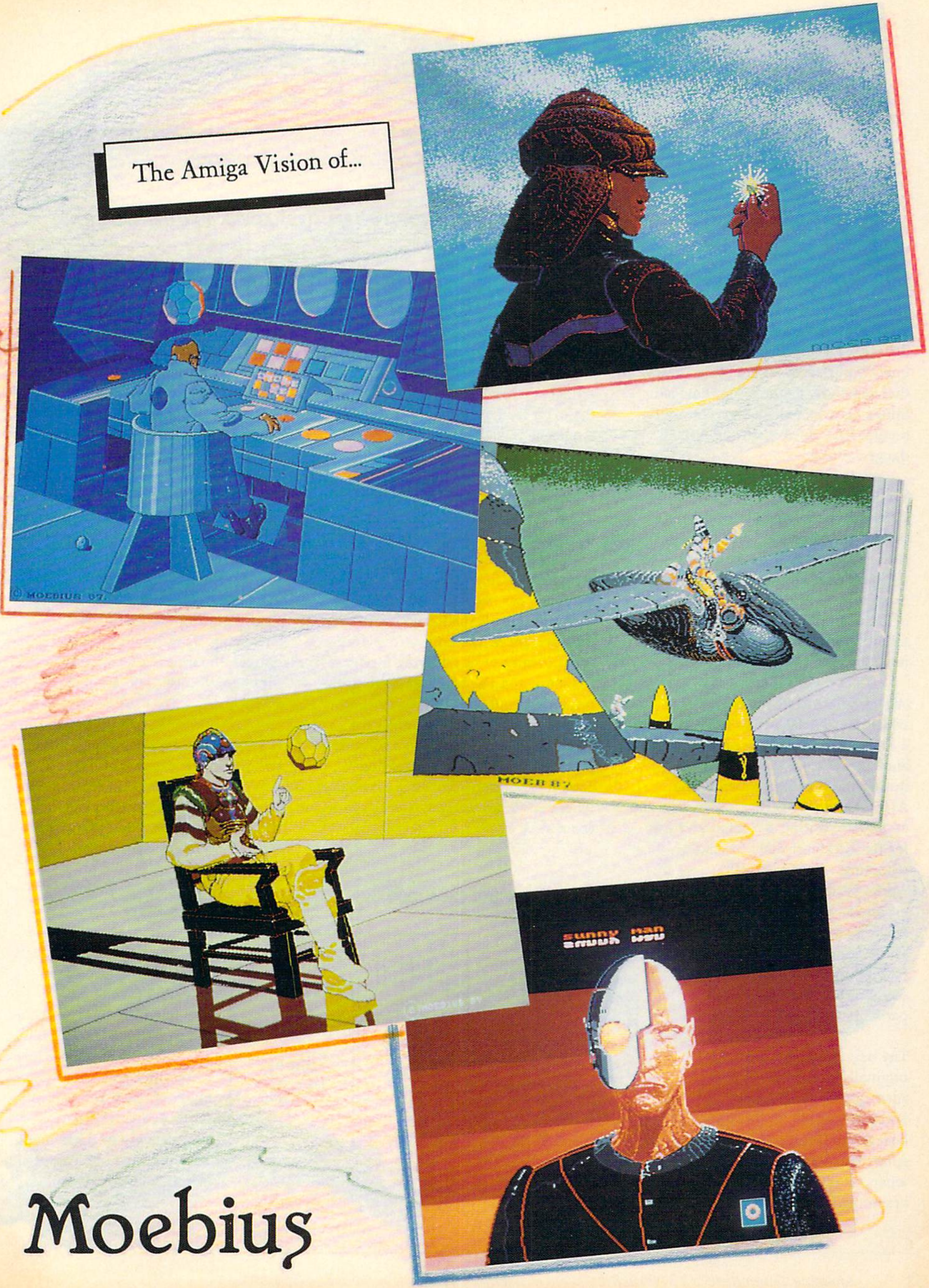
***AC:*** *What are your plans for the Amiga in the future?*

***MOEBIUS:*** My son is my guide in this area. With his help, I have only the intention to enjoy and explore. Discover what? That's the best part. It will be a surprise! Voila!

•AC•


J.V
TSHAI
75

## The Amiga Vision of...



# Moebius

# PAL Help

**by Perry Kivolowitz**

***"A simple procedure you can follow to improve your A1000's reliability..."***

Since the introduction of the Amiga 1000, rumors concerning its ability to drive expansion devices have frequently surfaced. A common theory is that somehow Commodore botched things up when choosing vendors for several PAL devices on the A1000 daughterboard.

While it is true that PALs from different vendors exhibit different drive characteristics, I have found a simpler, more realistic explanation of the problems some A1000s have with expansion products. In this article, I'll detail a simple procedure you can follow to improve your A1000s reliability with expansion products.

We've all heard about instances in which one A1000 works properly with a brand *x* expansion peripheral, but another does not. At some point during the booting sequence, the A1000 hangs, or perhaps it crashes unexpectedly during a session and asks for the KickStart disk, instead of rebooting. Or the A1000 works perfectly with one expansion product, but fails when a second is installed. These are common problems relating to the A1000 daughterboard (if not to the particular expansion product).

The daughterboard was not originally part of the A1000. Also called the KickStart RAM Tower, the daughterboard was thrown in to help speed the A1000 to market when no reliable ROM-based operating system was available. With 30 chips, the daughterboard replaces what only 4 chips would have provided in a ROM-based A1000. Consequently, the A1000 is more heavily burdened by the extra load than an A500 or A2000 would be.

One popular theory maintains that you must change four PALs on the daughterboard. According to this theory, replacing the existing PALs with stronger drive capability PALs compensates for the heavier load. Evidence has shown, however, that changing four PALs is successful only fifty percent of the time. Considering the difficulty of desoldering a twenty-pin chip, and the likely damage to the printed circuitboard, a fifty-percent success rate may not be worth the effort or risk. Anyway, this solution is particularly ineffective in solving certain expansion problems, especially where 68020 accelerator boards are concerned.

The daughterboard is connected to the motherboard by several one-inch metal pins. Two of the four PALs are locally grounded on the daughterboard itself, but the other two PALs are grounded on the motherboard, after traveling around the surface of the daughterboard and through the one-inch pins. Because of the poor grounding, these PALs pick up a lot of noise, causing interference for signals coming out of the expansion port. Providing better grounding for these PALs dramatically reduces the noise coming through the expansion port, and is much simpler than replacing the PALs. Anyway, if enhancing the grounding of the daughterboard fails to solve your problem, you can still try replacing the PALs.

If you do not experience any difficulties with your expansion product, you don't need to make this change. (In other words, if it's not broken, don't fix it.) Otherwise, you may wish to try the following change with your A1000.

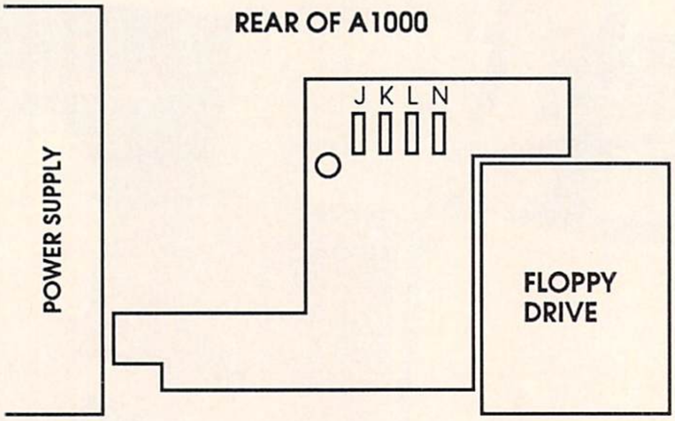


*Figure One*

**Alternatives**
You may wish to consider buying one of the KickStart ROM plug-in products available for the A1000, removing the daughterboard entirely (replacing it with what should have been there in the first place is a reasonable approach to this problem). However, these products are expensive and may produce unpleasant side effects for your A1000.

*Warning!*
*Neither ASDG Incorporated nor the publisher of this article can accept any liability for damage which may result from performing the change described below. This procedure is presented as a public service, and no warranties are expressed or implied. The procedure is safe and effective to the best of our knowledge.*

*Performing this procedure will, of course, void your warranty from CBM.*

**How It's Done**
Here's what you'll need:

- Phillips head screwdriver
- Flat head screwdriver
- Pair of needle-nose pliers
- Soldering iron
- Solder
- Insulated wire
- Wire stripper

Start by cutting four lengths of wire, each .85 inches long. Strip off 1/8 inch of insulation from each end. This should leave a bare section of wire .6 inches long. Using the needle-nose pliers, make a right angle bend at each end of the exposed wire.

Now let's remove the cover of your A1000. Disconnect the power, then remove all attached cables. Invert the A1000 and place it on a clean surface.

Locate the five recessed screws and use the phillips head screwdriver to remove them. Then slowly turn your A1000 right-side up. As you do so, the five screws will fall. Collect these and put them aside.

With your A1000 upright, turn the machine so that the right side (with the 86-pin expansion port) faces you. Locate the slight indentations on the side of your A1000 in the plastic bottom cover. There will be one in the bottom cover just above mouse port 2, and one an inch from the rear of the cabinet.

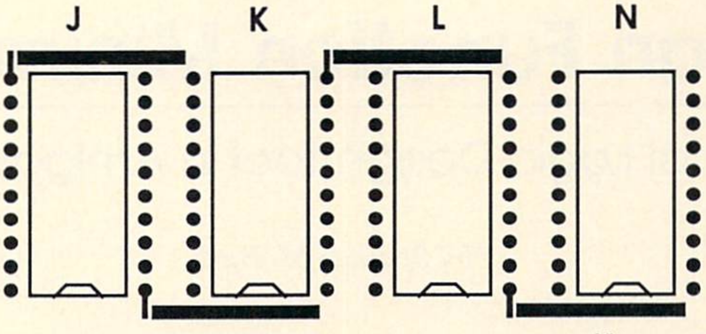


*Figure Two: Front of A1000*

Using the flat blade screwdriver, gently pry the top cover away from the A1000 by inserting the flat blade screwdriver into the recess. You are trying to pull a tab on the inside of the top cover out from its catching place in the bottom cover.

After freeing the right side of the top cover, work the left side free, so that the top cover can be removed entirely. You should now see the metal RF shield covering most of the inside of your A1000. Use the phillips screwdriver to remove the three screws along the power supply, one screw on the floppy disk drive, two screws along the front of the shield, and eight screws along the rear of it. Use your needle-nose pliers to pluck the screws out of the front and rear of the shield, since these are recessed beyond easy reach. Put these screws aside.

Before attempting to remove the metal shield, locate the four twist tabs securing the electrical contact between the ground of the motherboard and the shield. Two of these are located in the front of the A1000, one on the rear, and one on the right side. Using your needle-nose pliers, gently straighten the side and rear tabs so that the metal shield can be removed.

What you see should resemble Figure 1. Identify the daughterboard; it is the big green printed circuit board sitting one inch above the motherboard. Do not remove it.

Using Figure 1, locate PALs marked J, K, L, & N.

Using Figure 2, locate the pins on the top of J, K, L, & N. Connect the top left hand pins from J to K. Do the same from L to N. Using the remaining two wires, connect the bottom right pins of J & K, and connect the bottom right pins of L & N.

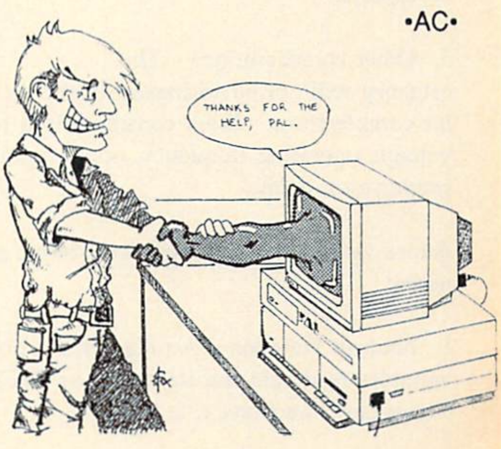
*Note: As you solder, the heat will be transmitted through the daughterboard to the actual PAL on the other side. To prevent damage to the PALs, be sure to solder quickly and accurately.*

The resulting connections should be identical to those shown in Figure 2. The new wires connect the ground and power pins of the better grounded PALs to the same pins on the poorly grounded PALs.

At this point, you may carefully connect the video and power cords to your A1000, as well as the expansion products you were having difficulty with. Be careful not to let anything fall into the open A1000. When you're done, reassemble the Amiga.

Your problems will probably have disappeared. If they persist, you may consider replacing the four PALs (J, K, L, & N), or eliminating the KickStart RAM Tower entirely. If your problems are not resolved, it's possible that your expansion products are at fault.

•AC•


THANKS FOR THE HELP, PAL.
PAL

# Boolean Function Minimization

## A Digital Logic Design Tool in AmigaBASIC™

*by Stephen M. Hart*

Of all the diverse fields in electronics, perhaps the most widely implemented is combinational (digital) logic design. The applications of this technology are as limitless as the imagination of the designer who applies it. However, before a digital design process can be pursued, the designer must carefully consider several constraints:

1. Size – How much space can the circuit occupy? This category directly correlates to the number of integrated circuit (IC) packages utilized in the design.

2. Cost – How much money will the design require? Does the cost justify the function of the circuit? Again, the number of ICs directly correlates to this category.

3. Fan-in – This is the number of inputs to a particular logic gate. Generally, the greater the fan-in, the greater the cost.

4. Fan-out – This is the number of logic gate inputs which can be driven by the output of a single gate. Serious problems may result if the design forces a gate to exceed its fan-out. The fan-out of a gate can be accurately calculated; however, for our purposes, a general rule of eight to ten per gate is acceptable.

5. Other considerations – This category will not be addressed in the article, but is included for completeness. Other considerations include operating voltage, operating frequency, operating temperature, and propagation delay.

Before we attempt a design, let's define a few elementary terms:

1. Boolean Function – An algebraic expression in which variables represent the state (ON or OFF) of the digital circuit (see Functions 1, 2, and 3).

2. Truth Table – A tabular representation of all possible inputs to a logic circuit, and the corresponding outputs (see Table 1).

3. NAND-NAND Realization – A Boolean function that can be realized (implemented) with two levels of NAND gates (see Figure 1).

4. Minterms – Decimal integer equivalents of the binary inputs to a logic circuit which yield 'high' or '1' outputs (see Table 1).

5. Don't Cares - Decimal integer equivalents of the binary inputs to a logic circuit which are independent of or irrelevant to the function of the circuit.

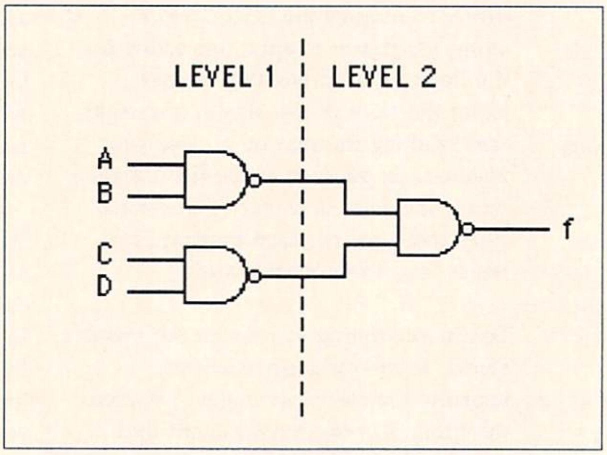


**Figure 1:** *2-Level NAND-NAND Circuit*

6. Literal – Any variable or negated variable within a Boolean expression.

7. Prime Implicants – Boolean terms which cannot be combined with any other term(s) to produce a new term with fewer literals.

8. Essential Prime Implicants – Boolean terms which must be present in the circuit implementation.

9. Nonessential Prime Implicants – All prime implicants which are not essential prime implicants. Nonessential prime implicants can be reduced to a minimum set required to implement the function.

With the above constraints and definitions in mind, let's step through the design of a simple combinational circuit. The truth table (see Table 1) illustrates a circuit with four possible inputs (A B C D) and one output (f). A "1"

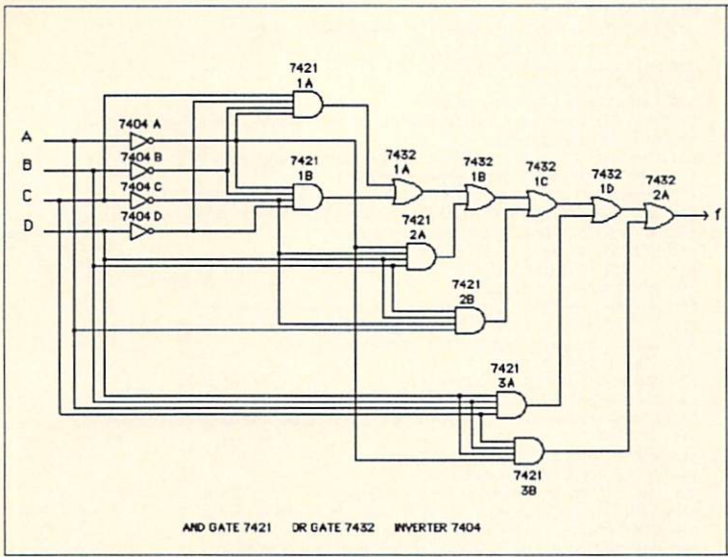


**Figure 2:** *Non-Minimized Realization of Table 1*

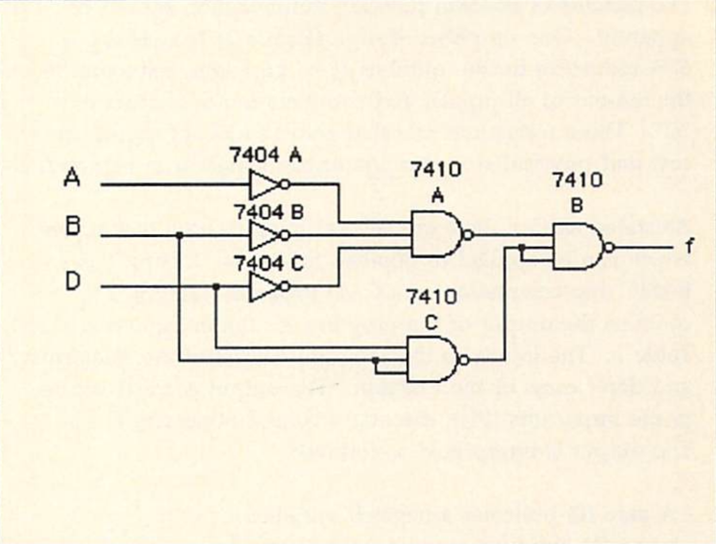


**Figure 3:** *Minimized Realization of Table 1*

indicates a logical "high" or "on" state, and a "0" represents a logical "low" or "off" state. The truth table can be directly transformed into a Boolean function representation:

f = A'B'C'D + A'B'C D' + A'B C'D + A'B C D + A B C'D + A B C D (Function 1)

We can now transform this equation into its combinational logic representation (see Figure 2). As illustrated in Figure 2, this implementation requires:

| **MINTERMS** | **INPUTS** | **OUTPUT** |
|---|---|---|
| | **A B C D** | **f** |
| 0 | 0 0 0 0 | 1 |
| | 0 0 0 1 | 0 |
| 2 | 0 0 1 0 | 1 |
| | 0 0 1 1 | 0 |
| | 0 1 0 0 | 0 |
| 5 | 0 1 0 1 | 1 |
| | 0 1 1 0 | 0 |
| 7 | 0 1 1 1 | 1 |
| | 1 0 0 0 | 0 |
| | 1 0 0 1 | 0 |
| | 1 0 1 0 | 0 |
| | 1 0 1 1 | 0 |
| | 1 1 0 0 | 0 |
| 13 | 1 1 0 1 | 1 |
| | 1 1 1 0 | 0 |
| 15 | 1 1 1 1 | 1 |

**Table 1**– *Combinational Logic Design Truth Table*

4 inputs (A B C D)
1 7404 hex inverter package
3 7421 dual 4-input positive-AND gate packages
2 7432 quadruple 2-input positive-OR gate packages

This yields a total requirement of 6 IC packages. Also notice that input "B" is required to drive 5 separate logic gate inputs. This requirement could present a problem if the quality of input "B" is poor. We could alleviate the possibility of this problem by passing input "B" through a line driver (another gate).

What can be done to improve our design? There are many Boolean function minimization techniques which can be applied to our original equation (Function 1). These include Boolean algebra, Karnaugh mapping, the Quine-McCluskey Method, and computer-aided design (CAD). Application of any of these techniques will yield the following simplified Boolean function:

f = A'B'C' + B D (Function 2)

An additional design procedure, double negation of the function, will transform our reduced function into its NAND-NAND realization:

f = ( (A'B'C')'(B D)' )' (Function 3)

The schematic for our reduced function is shown in Figure 3. This simplified design requires:

3 inputs (A B D).
1 7404 hex inverter package.
1 7410 triple 3-input positive-NAND gate package. I I

*(continued)*

The benefits of Boolean function minimization should be apparent. Our simplified design (Figure 3) has yielded a 67% reduction in the number of IC packages, reduction in the fan-out of all inputs, and complete removal of input "C". These reductions translate into significant savings in cost and physical size over the implementation in Figure 2.

As stated earlier, there are several minimization techniques which can be applied to Boolean functions. Listing 1 is a BASIC implementation of a CAD program. Listing 2 contains the output of the program for the design criteria of Table 1. The inputs to the program consist of the minterms and don't cares of the function. The output consists of the prime implicants (PIs), essential PIs, and nonessential PIs. The output is interpreted as follows:

- A zero (0) indicates a negated variable.
-A one (1) indicates a non-negated variable.
-A dash (-) indicates an omitted variable.

The program was developed on an Amiga 1000, but should be portable to most BASIC interpreters.

In conclusion, the design considerations, general terms, and CAD program presented here should start the reader on the right track to understanding and implementing low-cost, well-designed, digital logic circuits.

### Listing One
### CAD Boolean Function Minimizer

```
10 REM BOOLEAN FUNCTION MINIMIZER BY STEPHEN M. HART
20 CLS:PRINT:CLEAR ,25000:CLEAR ,50000&
30 DIM INTRM(100),MNTRM(100),BIN$(100,8),PI$(100,8),ESPI$(100,8)
40 DIM NSPI$(100,8),TEMP$(1,8),UN$(100,8),MNB$(100,8)
50 MN--1:PRINT:PRINT"ENTER MINTERMS - TERMINATE WITH NEGATIVE NUMBER"
60 INPUT NM
65 IF NM<0 AND MN<0 THEN PRINT"NO MINTERMS ENTERED!":GOTO 50
70 IF NM<0 THEN GOTO 100
80 MN-MN+1:IF MN>100 THEN PRINT"TOO MANY TERMS - 100 MAXIMUM":GOTO 800
90 MNTRM(MN)-NM:INTRM(MN)-NM:GOTO 60
100 INNM-MN:PRINT:PRINT"ENTER DON'T CARES - TERMINATE WITH NEGATIVE
NUMBER"
110 INPUT NM:IF NM<0 THEN 140
120 INNM-INNM+1:IF INNM>100 THEN PRINT"TOO MANY TERMS - 100
MAXIMUM":GOTO 800
130 INTRM(INNM)-NM:GOTO 110
140 REM PUT TERMS IN ASCENDING ORDER
150 FOR X=0 TO INNM-1:FOR Y-X+1 TO INNM
160 IF INTRM(X)<INTRM(Y) THEN 180
170 A-INTRM(Y):INTRM(Y)-INTRM(X):INTRM(X)-A
180 NEXT Y:NEXT X
190 ORDER-INT(LOG(INTRM(INNM))/LOG(2))
200 FOR X=0 TO MN-1:FOR Y-X+1 TO MN
210 IF MNTRM(X)<MNTRM(Y) THEN 230
220 A-MNTRM(Y):MNTRM(Y)-MNTRM(X):MNTRM(X)-A
230 NEXT Y:NEXT X
235 FOR X=0 TO INNM:FOR Y-0 TO ORDER:BIN$(X,Y)-"0"
237 IF X<-MN THEN MNB$(X,Y)-"0"
239 NEXT Y:NEXT X
240 FOR X=0 TO INNM:CON-INTRM(X)
245 IF CON-0 THEN 260
250 BIT-INT(LOG(CON)/LOG(2)):CV-ORDER-BIT:BIN$(X,CV)-"1"
255 CON-CON-2^BIT:GOTO 245
260 NEXT X
270 FOR X=0 TO MN:CON=MNTRM(X)
275 IF CON=0 THEN 290
280 BIT-INT(LOG(CON)/LOG(2)):CV-ORDER-BIT:MNB$(X,CV)-"1"
285 CON-CON-2^BIT:GOTO 275
290 NEXT X
300 REM PRODUCE THE PRIME IMPLICANTS
310 PINUM--1:FOR X=0 TO ORDER:TEMP$(0,X)-"0":NEXT X:X=0
320 Y-INNM
330 LG-INTRM(X) AND INTRM(Y):IF LG-INTRM(X) THEN 370
340 Y-Y-1:IF Y>-1 THEN 330
350 X-X+1:IF X<INNM THEN 320
360 GOTO 545
370 C$-"F":FOR V - 0 TO ORDER
380 TEMP$(0,V)-BIN$(X,V):IF BIN$(X,V)<>BIN$(Y,V) THEN TEMP$(0,V)-"-"
390 NEXT V:MCH--1:PSB-0
400 FOR A-0 TO ORDER:IF TEMP$(0,A)-"-"THEN PSB-PSB+1
410 NEXT A:IF PSB-0 THEN MCH-1:GOTO 460
420 CAN-2^PSB:FOR D-0 TO INNM:SAME--1:FOR A=0 TO ORDER
430 IF (TEMP$(0,A)-BIN$(D,A)) OR (TEMP$(0,A)-"-") THEN SAME-SAME+1
440 NEXT A:IF SAME-ORDER THEN CAN-CAN-1
450 NEXT D:IF CAN=0 THEN MCH-1
460 IF MCH<>1 THEN 340
470 IF PINUM--1 THEN 520
480 FOR C-0 TO PINUM:C$-"T":SAME--1:FOR D-0 TO ORDER
490 IF (TEMP$(0,D)-PI$(C,D)) OR (PI$(C,D)-"-") THEN SAME-SAME+1
500 NEXT D:IF SAME-ORDER THEN C$-"F":C-PINUM
510 NEXT C:GOTO 530
520 PINUM=0:FOR C-0 TO ORDER:PI$(0,C)-TEMP$(0,C):NEXT C
530 IF C$-"F" THEN 340
535 PINUM=PINUM+1:FOR V-0 TO ORDER:PI$(PINUM,V)-TEMP$(0,V):NEXT V
540 GOTO 340
545 REM OUTPUT THE PRIME IMPLICANTS
550 PRINT"PRIME IMPLICANTS:"
555 FOR V-0 TO PINUM:FOR T-0 TO ORDER
560 PRINT PI$(V,T);:NEXT T:PRINT:NEXT V:PRINT
570 REM PRODUCE ESSENTIAL PRIME IMPLICANTS
580 FOR X=0 TO PINUM:SAV(X)-0:NEXT X:NSAV=-1:ESNM--1
590 FOR X=0 TO MN:CNT=0:FOR Y=0 TO PINUM:MCH=-1:FOR P=0 TO ORDER
600 IF (PI$(Y,P)-MNB$(X,P)) OR (PI$(Y,P)-"-") THEN MCH-MCH+1
610 NEXT P:IF MCH-ORDER THEN CNT-CNT+1:HOLD-Y
620 NEXT Y:IF CNT<>1 THEN 660
630 IF SAV(HOLD)<>0 THEN 670
640 SAV(HOLD)-1:ESNM-ESNM+1
650 FOR Z-0 TO ORDER:ESPI$(ESNM,Z)-PI$(HOLD,Z):NEXT Z
660 NSAV=NSAV+1:FOR V-0 TO ORDER:UN$(NSAV,V)-MNB$(X,V):NEXT V
670 NEXT X
680 NON--1:FOR X-0 TO PINUM:IF SAV(X)<>0 THEN 700
690 NON=NON+1:FOR Y-0 TO ORDER:NSPI$(NON,Y)-PI$(X,Y):NEXT Y
700 NEXT X
710 REM OUTPUT ESSENTIAL PRIME IMPLICANTS
720 PRINT"ESSENTIAL PRIME IMPLICANTS:"
730 FOR V-0 TO ESNM:FOR T-0 TO ORDER:PRINT ESPI$(V,T);
740 NEXT T:PRINT:NEXT V:PRINT
750 REM OUTPUT NONESSENTIAL PRIME IMPLICANTS
760 IF NON=-1 THEN PRINT">>> NO NONESENTIAL PRIME IMPLICANTS <<<":GOTO
800
770 IF ESNM--1 THEN PRINT"THE FUNCTION IS CYCLIC":PRINT
780 PRINT"NONESSENTIAL PRIME IMPLICANTS:"
790 FOR X-0 TO NON:FOR Y-0 TO ORDER:PRINT NSPI$(X,Y);:NEXT Y
795 PRINT:NEXT X
800 END
```

### Listing Two
### CAD Minimization of Table 1

```
ENTER MINTERMS - TERMINATE WITH NEGATIVE NUMBER

0
2
5
7
13
15
-1

ENTER DON'T CARES - TERMINATE WITH NEGATIVE NUMBER
-1
PRIME IMPLICANTS:
00-0
-1-1

ESSENTIAL PRIME IMPLICANTS:
00-0
-1-1


>>> NO NONESSENTIAL PRIME IMPLICAMTS <<<
```

•AC•

# AmigaNotes

*by Rick Rae*
CIS (76703,4253)

## Modify your A1000 to have a software-switchable output filter, just like the A500 and A2000.

This month's column is devoted to a neat little hack for the Amiga 1000. With only five components that total under three dollars, you can now modify your A1000 to have a software-switchable output filter, just like the A500 and A2000.

Do I have your attention? Read on!

### AMIGA'S FILTER BYPASS OPERATION

I won't try to convince you that I was the first person to investigate bypassing the Amiga's output filter. I never felt comfortable about running a column on it, though; once done, it couldn't be undone, unless you mounted a switch somewhere on the case. Although this is fine for the private hacker, I wanted to present something a bit "cleaner" here.

When Commodore released the A500 and A2000 with their software-controlled filters, the solution became obvious: develop a modification which looked exactly like the factory approach. After switching over to this new circuitry and testing it for a few months, I'm ready to pass the details along to AC readers.

If you're going to be crawling around inside your Amiga, I want the changes to be as simple, foolproof, and easily installed as possible. This design fits those criteria. Absolutely no trace cuts or modifications are required. No wires are soldered directly to ICs or other active devices. You can later remove the additional circuitry to restore your machine to its original state. Still, I am obliged to make a disclaimer:

**WARNING:** *Although this is a very simple project, is requires disassembly and direct soldered connections to your Amiga. It will also void your warranty if your machine is new. In addition, the 4066 IC used is CMOS, and is therefore relatively sensitive to damage from static electricity while not wired into a circuit. If you are not confident about your electronic assembly skills, I strongly urge you to obtain assistance. Neither* Amazing Computing *nor I will be responsible for any damage resulting from attempting this modification.*

Are you still interested? Then read through the column at least once, pick up the items in the parts list, and dig in!

### PREPARING YOUR AMIGA FOR INSTALLATION

The first step, of course, is to get inside your Amiga. To begin, turn off the power and gently flip the machine over onto a soft cloth. (You might want to disconnect all the cables first.) You will see five screws recessed in the bottom housing. Remove only these five screws.

Now flip the machine back over and pull the cover off. The top snaps into place, so it won't lift right off. Instead, you'll have to apply a slight inward pressure to the bottom near the rear while pulling outward slightly on the top. The rear corners should pop loose, at which point you will be able to lift off the cover and set it aside. (While you have the hood off, be sure to admire the embossed names of the Amiga team members.)

Under the plastic cover, you'll see the metal shield which keeps the FCC happy. This one has a lot of screws, so be sure to get them all. There are eight along the back edge, three along the left side, two along the front, and one on the right at the disk drive. Be careful not to drop any screws down into the machine.

Once the screws are out, you must twist several tabs before you can remove the shield. This is easiest when done with needle-nose pliers, although you may be able to use a small set of standard pliers if you work carefully.

Now gently lift the shield and set it aside. The inside of your Amiga should look something like mine (Figure 1). The large "S" shaped board is the WCS daughterboard. (Notice the large "L" shaped board in Figure 1. This is my Insider 1 meg RAM board from Michigan Software. I heartily recommend this card to anyone who needs extra memory; mine was the eleventh off the assembly line, and I've never had any problems with it. This board was reviewed in AC V2.12 if you'd like to know more.)

*(continued)*

The area in which you'll be working is inconveniently located under the WCS daughterboard. Remove the three screws holding the daughterboard in place, then slip your fingers under the edges and gently rock it up and off the motherboard pins. Work slowly and carefully, prying up alternate corners a bit at a time. Once the daughterboard comes free, set it aside.

Notice that one of the cover support posts passed through a hole in the daughterboard. If you look just behind this post on the motherboard, you'll see U5G; this is the area where all but one of the connections will be made.

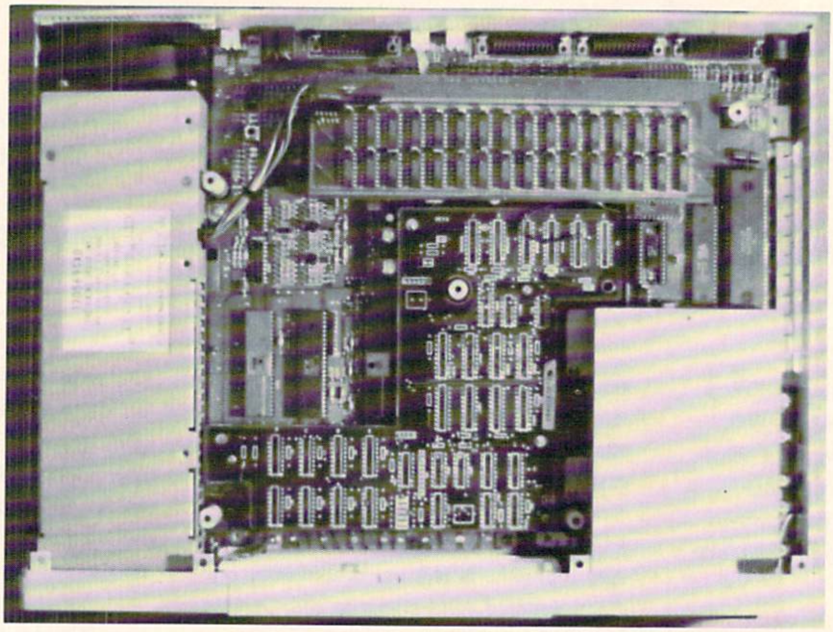


**Figure 1:** *My Amiga, with her cover removed, is ready for the filter BYPASS operation. The long box to the left is the power supply, with the fan behind; the box to the right is the floppy drive. You'll be working in an area under the WCS daughterboard, just above and to the right of the three large custom chips.*

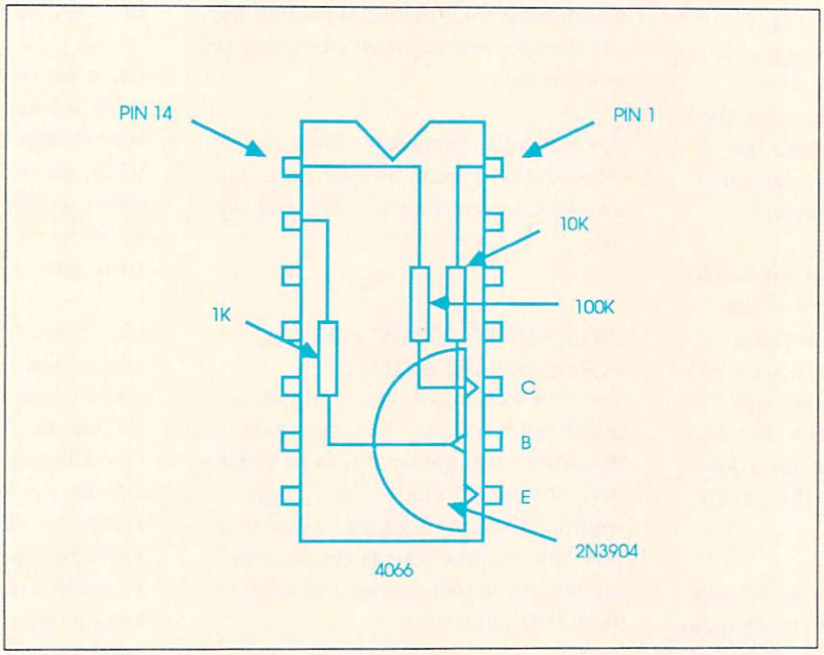


Figure 2: A diagram of how I mounted all the components on the 4066 IC. Notice that this view is from the bottom of the IC, so all pin numbers are reversed.

## BUILDING THE CIRCUIT

Now you're ready to start wiring. Since this is a very straightforward modification, you can use just about any construction method you'd like. If you're into overkill, you could make up a little PC board, but most people would probably be satisfied with a small piece of perfboard and an IC socket. What I will describe is an even simpler approach, called "dead bug" style.

Dead bug modifications are performed by mounting the ICs upside down (hence the name) on the PC board. Components and wire connections are made directly to the pins of the IC. The advantage of this approach is simplicity: it can be done in minimal time with no extra parts required. The disadvantage? The IC is hard to replace if it goes bad, or if you damage it during installation.

Begin by tying pins 2, 7, 10, 11, 12, and 13 of the IC together; also tie pins 5 and 6 together. Next come the other components. The layout is up to you, but all of them can be mounted neatly on the IC if you plan carefully. The 100K resistor goes between V+ (which is on pin 14) and pin 5 or 6, and can be mounted flat against the underside of the 4066. I mounted the 2N3904's collector and emitter directly to pins 5 and 7, respectively, bending the base lead back toward the center of the IC. The 1K resistor goes between the base and V- (pin 7), again flat against the IC. The 10K resistor connects to the transistor's base, with the other end connected to pin 1. Pin 1 has no effect on the circuit's operation; it is used merely as a tie point to avoid a "flying" component. Figure 2 is a bottom view of this parts layout.

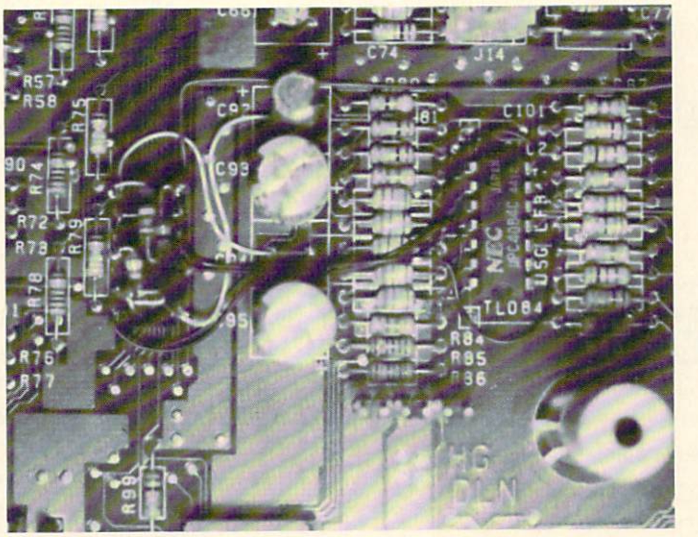


**Figure 3**: *The installed filter BYPASS. Notice the 4066, mounted upside-down, in the left of the picture. The TL084 (U5G) is in the center.*

Now that the components are mounted, it's time to install the circuit in the Amiga. Apply a piece of double-side foam tape to the top of the 4066 IC and fasten it, pins up, to your Amiga's motherboard; try to place it as close to U5G as possible. To the left of U5G is a group of capacitors and, beyond that, an area with several traces and no components. This is where I installed my modification; see Figure 3.

Once you have the IC mounted, you can begin wiring it to the motherboard; see Figure 4 for a diagram of the connection points. Notice that there are two pads marked by silkscreened squares, one in front and the other behind U5G; these are intermediate signal outputs from the Amiga's sound circuitry. These must be connected to the inputs of the 4066 IC; you will wire the front pad to pin 8, the rear pad to pin 4.

Now look at the row of components to the right of U5G. In the middle of this row, side by side, are two capacitors (they look like resistors) color-coded RED-RED-BROWN. These are the only components in this row with that code, so you can't go wrong. The left end of each capacitor is an intermediate signal input, and this is where the signal returns from our circuit. The front capacitor's left lead will be connected to pin 9, the rear one to pin 3.

Power supply voltages are available from the group of electrolytic capacitors to the left of U5G (and just to the right of where I installed my circuit). In my Amiga, "C94" was missing; it may exist in your machine. In either case, the right pin supplies the positive voltage to be used, and should be connected to pin 14 on the 4066.

Similarly, the right pin on C92 (the rearmost one) provides the negative voltage, and should be connected to pin 7.

The only other connection to be made is the control line. This one comes from the area near U6K; look to the right and back a bit. To the left of this IC are a pair of resistors; you want the front lead of R61, the left resistor. Connect this lead to the end of the 10K resistor connected to pin 1 on the 4066.

Got it? Double check your wiring before going further. Be absolutely sure of your connections before applying power; once you flip the switch, it's too late to back out.

Leaving the covers off temporarily, reconnect all the Amiga's cables and boot the system. Be sure to listen for the series of tones played before KickStart is loaded. If you don't hear them, or if they sound much "brighter" than normal, something is wrong. Immediately power down and recheck your wiring.

Once your Amiga is running, try any audio program which uses both channels to verify that things are still working properly. You should hear

*(continued)*

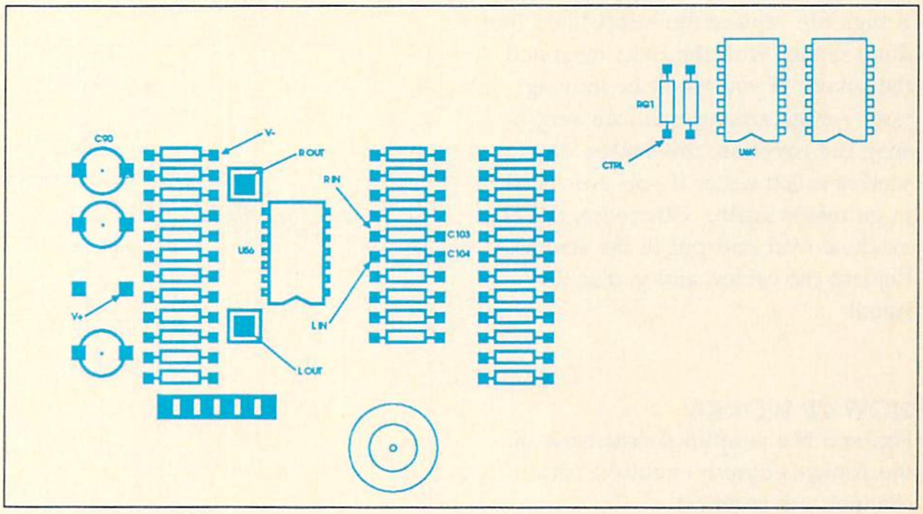


**Figure 4**: *This diagram shows the components on the Amiga motherboard around the audio output stage. The arrows mark the various connection points for the modification.*

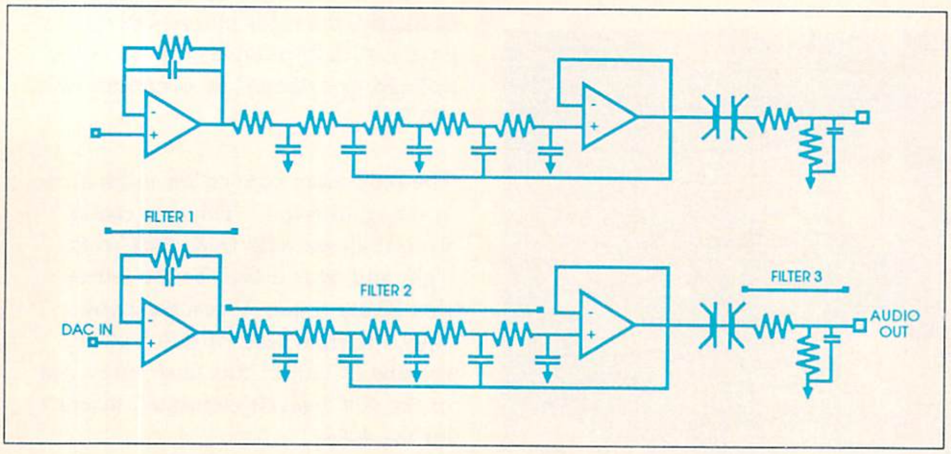


**Figure 5**: *A simplified schematic of the Amiga's audio output circuitry, showing the three filter sections.*

sound quality exactly like what you've heard since you bought your machine. Now you need to flip the state of the power LED. I suggest FILT.ARC in DL15 of the Amigaforum, but for testing purposes I've included a very simple Manx C program which you can compile. If you don't have a C compiler, you could write a Basic program to do the same job; just set the second bit of address $BFE001 to turn off the filter. Regardless of the method you use, run your audio program again once the power light is dimmed; you should be treated to a much brighter output, indicating that the filters are bypassed.

If everything seems okay, power your Amiga off, replace the metalshield (but don't bother with the twist tabs) and the cover. If you won't be moving your Amiga around, you can simply snap the cover on; this makes disassembly much easier if you ever need to go inside again. Otherwise, flip the machine over and put in the screws. Replace the cables, and you're finished!

## HOW IT WORKS

Figure 5 is a simplified schematic of the Amiga's output circuitry; both channels are identical.

The output from the D/A converter, which is a series of stairstep voltages, is fed to one section of U5G, a quad opamp. This section is configured as an integrator, which is, in effect, a lowpass filter. The integrator's job is to smooth out the stairsteps, providing a smoother waveform.

The output from the integrator is fed to a second opamp, configured as a very sharp multi-pole lowpass filter; this is where most of the filtering action takes place. Notice the resistor/ capacitor ladder between the two opamps; these components, in conjunction with the second opamp section, comprise the filter.

After the second opamp, a pair of back-to-back electrolytics block the DC voltage present, a voltage divider reduces the maximum output level to something more usable by a stereo system, and an RC network provides a final stage of filtration.

The output of the first opamp section is extremely low impedance, which means it will tend to "swamp out" any signals from a high impedance source. Notice the feedback from the second opamp which forms the filter: the output loops back through several capacitors and resistors before making its way back to the input. Because of this, the impedance of any filtered signal reaching the input will roughly equal the value of the resistors (fairly high) plus the impedance of the capacitors (also fairly high).

If we make a connection from the output of the first opamp to the input of the second opamp, the low impedance of the first stage blocks the operation of the filter. Effectively, the second opamp becomes a simple voltage follower and is totally transparent in its operation. (Thanks go to New Wave Software, the developers of Dynamic Drums, for this technique. My method was a bit simpler, but their approach required no modifications to the Amiga motherboard, and was easily adapted to my auto-switching circuit.)

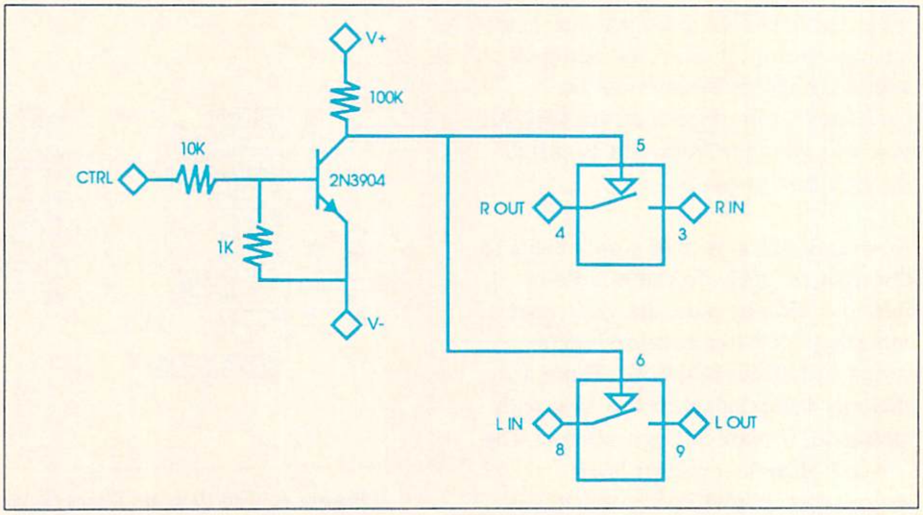


**Figure 6**: *The schematic for the filter BYPASS modification. Not shown are the power supply and ground pins, and the unused pins, all of which are tied to V-.*

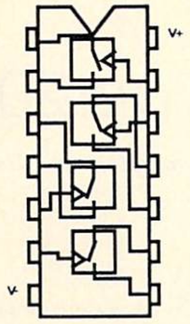


**Figure 7**: *The pinout for the 4066 quad analog switch.*

Now take a look at Figure 6, which is a schematic of the modification I've described. The 4066 is a quad analog switch and—you guessed it—we use two switches to connect the output of the first opamp to the input of the second for each channel.

The control signal is taken from the driver used for the power LED. Since the sense of the signal is backwards, a 2N3904 is used as an inverter to drive the control lines for the analog switches. (I had hoped to use one of the unused 4066 switch sections as an inverter; this would have made a one-component circuit! Unfortunately, the voltage swing at the LED control line is not sufficient to guarantee proper operation. The transistor, therefore, also acts as a level shifter and gain stage.)

Figure 7 is a pinout of the 4066. With the exception of pin 7, all the pins which are tied to V- are unused inputs or outputs; this prevents the excessive current drain which would otherwise occur if any of these pins drifted into their linear regions.

Although this method is not exactly the same as that used in the A500 and A2000, the audio results are effectively identical. Any program which switches the filter on the new machines should work properly with this circuit as well.

## OTHER FILTERS

You probably caught the reference to three filters, and may be wondering about going even farther. Although this is possible, it should be done with care.

The obvious step is to remove the final filter by clipping out C5 and C6 (located immediately adjacent to the audio output jacks). With the multi-pole filter enabled, you won't hear any difference to speak of, so this change doesn't impair normal operation. With the filter bypassed, you'll notice a considerably brighter top end. While this will make some signals sound better, others will sound worse because of aliasing distortion. This one will have to be your decision; I believe you'll generally like the sound better with the capacitors removed, but your ears are the ones that must be satisfied.

What about the integrator, the first stage immediately after the D/A converter? This "filter" can be bypassed by clipping C96 and C100. I don't recommend this change, however; with no filtration whatsoever, a digital system will tend to sound rather harsh and tiring. If you are interested in pursuing the highest possible audio quality, you could clip these two capacitors and use an external filter with a very sharp corner. I would urge the typical user, however, to leave the integrator alone.

That will do it for now. Have fun with your new "high fidelity" Amiga 1000, and I'll see you next month!

Nybbles, Rick

### *PARTS LIST*

1K Ohm 1/4 watt resistor; 10K Ohm 1/4 watt resistor; 100K Ohm 1/4 watt resistor; 4066 quad analog switch; 2N3904 NPN Transistor

Notes: A 4016 may be substituted for the 4066, with some degradation of performance. Wire-wrap wire is suggested for connections, but any small-gauge wire may be used. A small piece of double-sided foam tape will be required if you wish to mount the IC "dead bug" style.

•AC•

```
/*****************************************************\
* Program FILTER: Toggle state of Amiga audio filter. *
*******************************************************
* Note: This program operates directly on the Amiga's *
* hardware.  if anyone knows a system call for access *
* to this bit please let me know; I couldn't find it. *
\*****************************************************/

#include <hardware/cia.h>
/* Include CIA definition */

void main()

{ short int temp;

temp = ciaa.ciaddra;
/* Get current DDR status */
ciaa.ciaddra |= CIAF_LED;
/* Set LED line to output */
ciaa.ciapra ^= CIAF_LED;
/* Flip state of LED line */
ciaa.ciaddra = temp;
/* Restore old DDR status */

if(ciaa.ciapra & CIAF_LED)
  puts("Filter is bypassed.");
  else puts ("Filter is enabled.");

}
```

# Amiga 1000 Serial Port and MIDI Compatibility For Your Amiga 2000!

*By Lynn Ritter and Gary Rentz*

Have you been singing those Amiga 2000, MIDI incompatibility blues? I just might be able to help you change your tune and chase those blues away.

Did you make a gender bender with the five-volt circuit from AC V2.12 and find it didn't work? After a little investigating, I found a better solution. This article describes how to add a serial port to your A2000 that is the same as the one on your Amiga 1000, if you are into something other than MIDI.

As you know, the Amiga 2000's serial connector was changed to a male connector, and the +5 and –5 volts were dropped. That applies to the *outside* serial connector, but inside your Amiga 2000 is a different connector, with all the right things in all the right places on a user serial port header.

This internal serial connector is located on the main board right above (and about a half-inch from) the outside serial connector. Once you pop off the cover, find the outside connector and look inside at the same area to see a header with two rows of thirteen pins sticking up. The connector runs parallel to the back wall of the computer, and there are two cover plates through which you can run your cable if you don't want to connect the DB25 to the Amiga.

The internal connector is a 26-pin header which mates with a berg connector (usually referred to as a

**To Front of Amiga**

——————>

| DB 25 Pin # | | | | | DB 25 Pin # |
|---|---|---|---|---|---|
| 1 | Ground | X 1 | 2 X | -5 Volts | 14 |
| 2 | TX | X 3 | 4 X | Audio Out | 15 |
| 3 | RX | X 5 | 6 X | Audio In | 16 |
| 4 | RTS | X 7 | 8 X | Port Clock | 17 |
| 5 | CTS | X 9 | 10 X | Interrupt | 18 |
| 6 | DSR | X 11 | 12 X | * LED Power | 19 |
| 7 | Signal GND | X 13 | 14 X | DTR | 20 |
| 8 | Carrier Detect | X 15 | 16 X | + 5 Volts | 21 |
| 9 | ——— | X 17 | 18 X | * Ring Indicator | 22 |
| 10 | ——— | X 19 | 20 X | +12 Volts | 23 |
| 11 | ——— | X 21 | 22 X | Clock 3.58 Mhz | 24 |
| 12 | ——— | X 23 | 24 X | Buffered Reset | 25 |
| 13 | ——— | X 25 | 26 X | ——— | 26 |

* Not provided on Amiga 1000

2 X 13 —or however many pins are in one row). On an IBM, this is the very same connector as the one that appears on the serial cards, where a ribbon cable is brought out to the back as a serial port and a DB25 male connector waits to be connected to a female connector. With this ribbon cable from the PC, all signals and voltages are in place for your MIDI. You just need to connect a gender bender or cut off the male DB25 and solder on a female connector.

I was helping a friend hook up his Golden Hawk MIDI. Once we found the inner serial connector, I made a fast trip home and tore apart my nearly useless PC clone to get a couple of female DB25s.

We then started to hook them up. At first my friend was skeptical, saying "It can't be that easy; the pin-outs won't check, the signals will be in the wrong places!" I reminded him that the Amiga 2000 was supposed to be IBM compatible. That is why we were having trouble in the first place, remember? Well, all the signals we needed were right where they were supposed to be after being traced through the ribbon cable.

I soldered pins 1 through 4, 7, 14, 20, 21, and 23 between the two female DB25 connectors for a gender bender, and we had MIDI sound again. A regular gender bender from your local parts supplier will do nicely, but at 1 AM on Monday morning, it's hard to find an open store, so I did the soldering.

Now for the bad news! While those ribbon cables are common on PCs and compatibles with serial boards in them, they might not be easy to buy as single items. I was lucky enough to have a clone with a cable in it to cannibalize. You might not be so lucky. I called several places and could not find any cables for sale. If you can solder at all, any berg 2 X 13 connector with ribbon cable at least 6 inches long will do. If you can get the berg 2 X 13 connector (also called ribbon receptacle), you can hook up a ribbon-style, D-series female connector yourself. It is also possible to solder the ribbon cable, but that might not be easy. You can also use a berg 2 X 14 or longer, and cut the connector down with a pair of side cutters. That is what I did to get my PC cable back.

If you haven't seen the *Commodore/Amiga A500/A2000 Technical Reference Manual*, I have included a listing of the connections from the 26-pin header to the DB25 connector. If you connect all 18 connections to the DB25, you will have a normal Amiga 1000 serial port.

If you are not a "do-it-yourselfer," you can contact the Illinois Computer Cable company (800-323-2612) and ask for an "Amiga 1000 to Amiga 2000 serial adapter cable." When I talked to them, the price was estimated at about 12 to 14 dollars. The adapter will also have the desired female connector, so your old serial cables and MIDI equipment will plug right in.

It's nice to know that you can upgrade to an Amiga 2000 and still use your old peripherals. It is up to you how you make the connection. If you have IBM connections, you might be able to get a cable like the one I used and use a gender changer. If you like to tinker and can find the connectors, you can make your own cable. If you are into electronics, you can build a multi-purpose gender changer/voltage changer circuit and hook your MIDI to the external serial connector (as long as you have enough current to power your MIDI).

So turn those Amiga 2000, MIDI incompatiblity blues back into your favorite style of music, and keep the music flowing. Music is food for the soul, so eat hearty.

•AC•

F-BASIC™

A **FAST** Compiled Enhanced **BASIC** Language

☐ The beginning programmer will appreciate the simplicity and ease of use of the F-Basic™ system. An entire disk of sample programs and a comprehensive manual are included.

☐ The advanced programmer will appreciate the many enhanced features supported by F-Basic™:

- Extensive Control Structures
- Recursive SUBROUTINEs & FUNCTIONs
- LOCAL & GLOBAL Variables
- 32, 16, And 8 Bit INTEGERs
- Ultra Fast 9 Digit Floating Point Routines
- Powerful String Manipulation Facilities
- RECORD Structures And Pointers
- Easy High Level Access To AMIGA™ Screens, Windows, Menus, Sound, Speech, And Graphics
- Direct Access To ROM Kernel Routines
- INCLUDE And APPEND Separate Files
- SNOBOL4™ Like PATTERN Matching
- Direct Access To 68000 Processor Registers and Bitwise Operations

☐ The included FastCom™ compiler is a one pass optimizing compiler generating in-line 68000 machine code. Extensive Compile and Run Time Errors. FastCom™ compiles thousands and thousands of lines per minute because it is written in F-Basic™!

**A proven system.**

**ONLY $79.95 (U.S.)**
**complete**
**NOW SHIPPING!**
Send check or money order to:
**DELPHI NOETIC SYSTEMS, INC.**
P.O. Box 7722
Rapid City, SD 57709
For fastest delivery call **(605) 341-2580**

F-Basic™ and FastCom™ are registered trademarks of DNS, Inc. AMIGA™ is a registered trademark of Commodore/AMIGA, Inc. SNOBOL4™ is a registered trademark of Bell Telephone Laboratories.

# Electric Network Solutions

### "...practical routines for using matrix

### Introduction

Matrix algebra is a very powerful tool, particularly in the field of electrical engineering. It is common for both engineers and engineering students to look for tools to relieve them from repetitive calculations. Personal computers and engineering work stations perform these tedious operations with ease.

This article is the second in a series providing practical routines for using matrix algebra for engineering applications. A basic understanding of the principles set forth in the first article is needed to understand the applications provided. This article assumes that the reader has implemented the library functions supplied with the first article and is now ready to use them.

One of the most common uses for matrix algebra is in solving electric network problems. This article will show, by example, how to integrate the library functions with their applications in network analysis. You should have some working knowledge of electric networks and of Ohm's law.

All the example programs will include the matrix routines developed in the previous article. In the case of the BASIC examples, the routines are to be contained in the same file as the library matrix functions.

### Loop and Node Equations and Kirchhoff's Laws

The solutions to electric network problems in this article are based on the use of matrix algebra to solve loop and node equations. The network equations that will be formed into matrices are derived from Kirchhoff's Laws.

Kirchhoff developed two laws that apply to electric networks, a current law and a voltage law. The examples used to introduce these concepts will be based on DC voltage sources and purely resistive loads. However, the methods developed are not limited to these restrictions.

### Kirchhoff's Voltage Law

Kirchhoff's Voltage Law states that the algebraic summation of each branch voltage around a closed network loop will always equal zero. This means that by adding all voltage sources and all voltage loads in a network loop, you can account for all of the voltage potentials. In other words, the sum of the voltage rises are equal to the sum of the voltage drops around a closed loop of electrical elements.

Kirchhoff's Voltage Law demonstrates the principle of the conservation of energy. The energy in this example happens to be the voltage potentials (where voltage is work per unit of time). An example of this law is shown in the following three-element circuit.

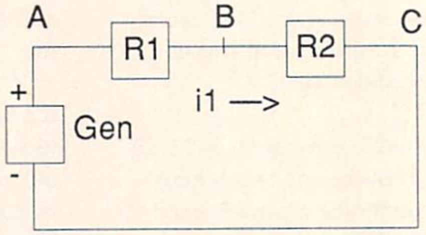


Figure 1

In this example there are three branch voltages:

Vac = the voltage across the generator; Vab = the voltage across resistive load R1; Vbc = the voltage across resistive load R2.

Kirchhoff's voltage law demands that the following equations be true for the circuit above.

Vac = Vab + Vbc or Vac - Vab - Vbc = 0

# the Matrix Way

by Robert Ellis

***algebra for engineering applications."***

Any change in voltage potential of the generator will be reflected in a change in the voltage potential across both of the load resistors. Because the voltage law always holds true, the voltage source can be either a DC or an AC source to satisfy the equations.

### Kirchhoff's Current Law

Kirchhoff's current law states that the algebraic sum of all branch currents entering and leaving a node must always equal zero. This law is a restatement of the law of the conservation of charge.

This law can be demonstrated with the circuit in the first example (Figure 1). Because the circuit is a single closed loop, there is only one loop current. This loop current is the same everywhere in the loop. If we take node B as our reference node, then the following equations are true:

Iab = Ibc     or     Iab - Ibc = 0

Because the equations are always presented with unknown values on the left side, one can easily set up the matrix equations for solution of the networks.

### Two-Loop Example Using Determinants

This example demonstrates the use of determinants in solving a simple network problem. The network contains a voltage source and three resistive loads. The configuration of the circuit is such that two distinct loops are formed. The object is to form two simultaneous equations with two unknown variables.

The general form of loop equations is the summation of voltage potentials around a loop (Kirchhoff's voltage law). This will result in the formation of equations where the values of the loads and voltage potentials are known. The next task is to find the loop currents which must exist to satisfy the law of the conservation of energy.

This form is shown in the following matrix:

Z11*I1 + Z12*I2 ... Z1i*Ii = V1
Z21*I2 + Z22*I2 ... Z2i*Ii = V2
..
Zj1*I1 + Zj2*I2 ... Zji*Ii = Vj

*Where*
Z = branch impedances (a known quantity);
V = source voltages (a known quantity)

Using Cramer's rule (see information box), each of the unknown currents becomes a ratio of two determinants. Therefore, if we want to solve for I1, the following ratio must be evaluated:

$$I1 = \frac{N1}{D}$$

where D = the determinant of the unknown variables and Ni = the determinant formed by substituting the resultant column matrix into the unknown column to be solved for.

A numerical example will demonstrate the operations described. The circuit in figure one is a simple two-loop circuit made up of resistive loads and a voltage source. To aid in setting up the equations and matrices required to solve for the currents, we will use a dummy voltage source in one loop.

Next, write out the loop equations for the two sections of the circuit. To write the first loop equation, find the voltage across each element in the loop.

24 volts = (4ohms * i1) + (6ohms * i1) - (6ohms * i2)

0 volts = (-6ohms * i1) + (12ohms * i2) + (6ohms * i2)

(continued)

INTERCHANGE ™

**Share objects between Sculpt3D, VideoScape 3D & Forms in Flight**

**NOW YOU CAN...**

- Use Sculpt 3D to ray–trace VideoScape 3D objects
- Do Forms in Flight animations on Sculpt 3D objects
- Create VideoScape 3D objects using the Sculpt 3D interface

**Full Intuition interface for all InterChange functions**

Object Disk #1 now available. Includes a Sculpt 3D font, plus lots more Sculpt 3D and VideoScape 3D objects!

InterChange master program plus Sculpt 3D and VideoScape 3D Conversion Modules, $49.95. Forms in Flight add–on Conversion Module, $19.95. InterChange Object Disk # 1, $19.95.

This product requires objects from Sculpt 3D and/or VideoScape 3D and/or Forms in Flight. It is not a standalone animation program.

To order, send check or money order. Please include $3.00 postage & handling. MA and WI residents add 5% sales tax. InterChange is a trademark of Syndesis. Sculpt 3D, VideoScape 3D and Forms in Flight are trademarks of Byte by Byte Corporation, Aegis Development and Micro Magic respectively.

**SYNDESIS**

20 WEST STREET
WILMINGTON, MASSACHUSETTS 01887
617–657–5585

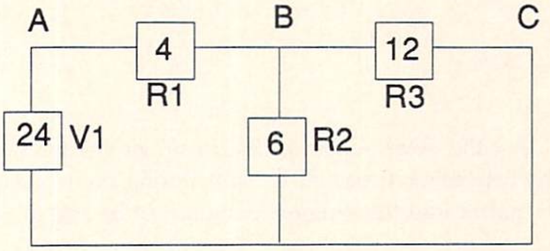


Figure 2

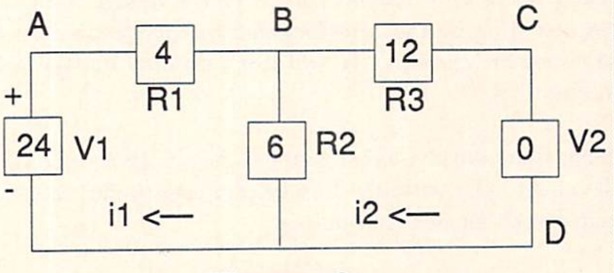


Figure 3

R1, R2, and R3 are resistors.
V1 is a 24 volt DC supply.
V2 is a 0 volt DC dummy supply.
i1 is the loop current for V1, R1, and R2.
i2 is the loop current for R3, R2
and dummy source V2.

Notice that each of the loops has a "current * resistance" term from the other loop. Notice also that this term is subtracted from the present loop's value. To see this more clearly, draw the loop current as a complete circle with a direction arrow on each side of the loop (bo* formed by included elements). Now you can see that the 6 ohm resistor in the center has both loop currents running through it, against each other. The illustration shows why this term is subtracted from the equation.

Reduce the equations above to the following form:

$$10i1 - 6i2 = 24$$
$$-6i1 + 18i2 = 0$$

Then place the equations into a matrix configuration. This new form will help you decide how to use matrix library routines to get the unknown variables (which are the loop currents i1 and i2).

$$\begin{vmatrix} i1 \\ i2 \end{vmatrix} \quad \begin{vmatrix} 10 & -6 \\ -6 & 18 \end{vmatrix} \quad = \quad \begin{vmatrix} 24 \\ 0 \end{vmatrix}$$

Now solve for the unknown current of i1, applying Cramer's rule in the following way. The first determinant of the square matrix (D) is found. Next substitute the result column matrix for the i1 column in the square matrix and find the determinant of the new square matrix (N). To obtain the actual loop current i1, scalar N is divided by scalar D. The final i1 loop current should be 3 amperes.

$$D = \begin{vmatrix} 10 & -6 \\ -6 & 18 \end{vmatrix}$$

determinant of D = (10 * 18) - (-6 * -6) = **144**

$$N = \begin{vmatrix} 24 & -6 \\ 0 & 18 \end{vmatrix}$$

determinant of N = (24 * 18) - (0 * -6) = **432**

**i1 = N / D = 432 / 144 = 3 amperes**

Although this simple example was not difficult to calculate, a real problem would be difficult and time consuming. With the tools provided, the matrix library routines do all the work. The following set of program statements (in AmigaBASIC) will perform the actual calculations for the example above.

```
a(0,0)   =   10
a(1,0)   =   -6
a(0,1)   =   -6
a(1,1)   =   18
```

```
    b(0,0)    =    24
    b(1,0)    =    0
    b(0,1)    =    -6
    b(1,1)    =    18

REM Solving for only one loop current, matrix cover-up
 REM occurs
CALL matdet(2,r1,b())
CALL matdet(2,d,a())
 r = r1/d
 PRINT "Loop current i1 = ";r
```

### Two-Node Example Using Determinant Method

This example also demonstrates the use of determinants to solve an electrical network. The network contains two current sources and three resistive loads. The circuit's configuration is such that there are two distinct nodes. The object is to form two simultaneous equations with the two unknown variables as the node voltages.

The general form of the node equations is the summation of the currents in the branches of the circuit (Kirchhoff's current law). Unlike the loop equation example, the values of the loads will be in admittance values, not in impedance values. This allows us to multiply the elements in the equations. The form of the node equations is shown in the following matrix:

Y11*V1 + Y12*V2 ... Y1i*Vi = I1
Y21*V2 + Y22*V2 ... Y2i*Vi = I2
..
Yj1*V1 + Yj2*V2 ... Yji*Vi = Ij

*where*
Y = branch admittance
( 1/Z ) which is the reciprocal of the branch impedance
I = source current

Again using Cramer's rule, determine each of the unknown branch voltages as a ratio of two determinants. To solve for the branch voltage V1, the following ratio must be evaluated:

V1 = N1/D

*where*
D = the determinant of the admittance matrix.
N = the determinant formed via the substitution of the resultant column matrix into the unknown column to be solved for.

Notice that the admittance matrix can be formed directly from an impedance matrix by taking the inverse of the impedance matrix. This will be useful to remember when performing both node and loop analysis on a circuit.

**Geodesic Publications**
*presents*
**Source Code Kits**
**for MANX® users**
**Powerful tools in C and Assembly**

**GEOTask™ I*** 27.00
CopperVBeam Interceptor for flicker free frames and colors. Replace the slow ROM graphics in your existing programs. Polygon Drawing with adjustable clipping, backdrop & SuperHAM. All Screen Formats, full and buffered video, IFF picture load and save.

**MIDISynergy™ I*** 37.00
Transparent interrupt driven Recorder and Sequencer. Tweeked! Mouse controlled modulation, pitch, tempo, and transpose. OpCode Format load & save. Loader converts SMUS into MIDI.

**MARS I*** Mindlight® Artist Realtime Software 27.00

**GeodesicNet™ II PD** 7.00

**Geodesic Publications**
P.O. Box 956068
Duluth, GA 30136
(404) 822-0566

MANX® Manx Software Systems
Mindlight® Visual Aural Animation
Georgia residents add 4% tax
*Includes color space art.

Now an example of the methods described above will be demonstrated on a circuit with real values in it.

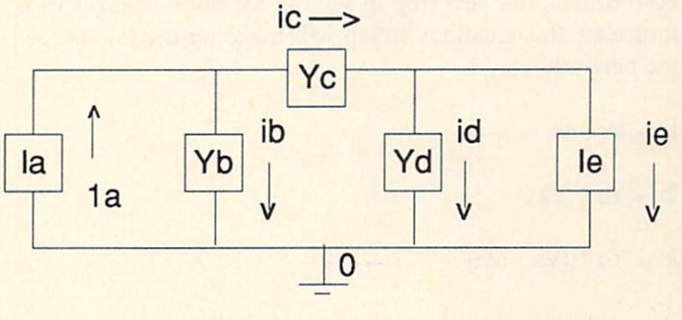


**Figure 4**

Ia = 3 amperes; Ie = 2 amperes

Yb = .5 mhos ( 2 ohms );
Yc = .125 mhos ( 8 ohms );
Yd = .25 mhos ( 4 ohms )

*(continued)*

IN-BOARD™

**PROVEN Internal Memory Expansion For The Amiga® 500 and 1000.**

Add up to 1.5 MB of Zero Wait-State Fast Memory for a total of 2.5 MB on your Amiga 500 and 2.0 MB on your Amiga 1000.

- Available in various user-expandable configurations from 0K to 1.5 MB with diagnostics included.
- Battery Backed Clock/Calendar standard on 1000 Version, optional on 500 Version.
- Fast, solderless, no-modification installation.
- Safely uses Amiga power in both U.S. and European models. (650 ma. max. 5vdc)
- Detailed installation instructions included on disk.
- Complete set of system utilities (auto-config, ram on/off, readclock, protected ram disk, more) with automatic install routines included.
- Factory technical support.
- Affordable! Only $279.00 for Inboard 500–0K.
- Satisfaction GUARANTEED, or your money back!

See your dealer or contact Spirit Technology.

INBOARD 500

INBOARD 1000

*DISK OFFER!*

Send $3.00 for a Demo Disk that shows just how easy it is to install your Spirit INBOARD. Disk is packed with many useful PD programs and utilities.

Specify A-500 or A-1000

220 West 2950 South
Salt Lake City, Utah 84115
(801) 485-4233
Major Credit Cards

**1-800-433-7572**

SPIRIT™ TECHNOLOGY

SPIRIT TECHNOLOGY CORPORATION

The arrows in Figure 4 show the direction of the flow of current in the circuit. The reference node is labeled O and is shown to be a ground potential. After the diagram has been drawn, the next step in solving the node voltages is to formulate the equations which will make up the matrix for the network.

$Ia = Ib + Ic$

$Ib = Yb * Va$

$Ic = Yc * (Va - Vb)$

$-Ie = Ic + Id$

$Id = Yd * Vd$

Collecting like terms in the above set of equations yields the following two equations:

$Ia = ((Yb + Yc) * Va) - (Yc * Vb)$

$-Ie = -(Yc * Va) + (Yc + Yd) * Vb)$

Now we have two equations in which the elements are composed of an impedance * a voltage. These two equations can be formed into a set of three matrices which will be voltage, admittance, and current. This is shown below.

$$\begin{array}{cccc} \textbf{Voltage} & \textbf{Admittance} & & \textbf{Current} \\ \begin{vmatrix} Va \\ Vb \end{vmatrix} & \begin{vmatrix} (Yb + Yc) & Yc \\ Yc & (Yc + Yd) \end{vmatrix} & = & \begin{vmatrix} Ia \\ Ie \end{vmatrix} \\ \begin{vmatrix} Va \\ Vb \end{vmatrix} & \begin{vmatrix} .625 & .125 \\ .125 & .375 \end{vmatrix} & = & \begin{vmatrix} 3 \\ 2 \end{vmatrix} \end{array}$$

To solve for one of the unknown voltages using the determinant method, we need to form the substitution matrix. To solve for Vb, we form the N2 matrix by substituting the second column of the admittance matrix with the resultant current matrix. The scalar result of the determinant for this newly formed matrix is then divided by the scalar result of the determinant of the full admittance matrix. The quotient will level us with the scalar value for the Vb node voltage (with respect to O the datum).

$$D = \begin{vmatrix} .625 & .125 \\ .125 & .375 \end{vmatrix}$$

determinant of D = (.625 * .375) - (.125 * .125)

**D = .21875**

---

$$N2 = \begin{vmatrix} .625 & 3 \\ .125 & 2 \end{vmatrix}$$

determinant of N2 = (.625 * 2) - (.125 * 3)

**N2 = .875**

---

**N2 / D = .875 / .21875 = 4 volts**

Again, the example used a small number of unknowns and equations so it would be easy to follow. The following set of program statements will perform the actual calculations.

```
a(0,0) = .625
a(1,0) = .125
a(0,1) = .125
a(1,1) = .375

b(0,0) = .625
b(1,0) = .125
b(0,1) = 3
b(1,1) = 2

REM Solving for only one node voltage, matrix cover-up
occurs
 CALL matdet(2,n,b())
 CALL matdet(2,d,a())
 r = n/d
PRINT "Loop current i1 = ";r
```

### Two-Loop Example Using Inverse Matrix Method

As outlined in the first article, there is a method which solves the unknown variables simultaneously. This method requires the inverse matrix function. The following outline uses the two-node network solution as an example of how this operation is set up and performed.

**Step I**

Set up the matrices from the two-node example so that the coefficients are separated from the unknown variables on the left side of the equation. The right side of the equation should contain the resultant column matrix. This yields the following form:

$$[A] * [X] = [K]$$

*where*

[A] is the coefficient matrix and contains the following values

$$\begin{vmatrix} 10 & -6 \\ -6 & 18 \end{vmatrix}$$

[X] is the unknown variable column matrix and contains the following variables for the circuit

$$\begin{vmatrix} i1 \\ i2 \end{vmatrix}$$

[K] is the resultant column matrix and contains the known voltages in the circuit

$$\begin{vmatrix} 24 \\ 0 \end{vmatrix}$$

# Something New Is Blowing Up a Storm!

## dbPROFESSIONAL

Bring the power of real database management, available for years in the IBM-PC environment, to your Commodore **AMIGA** computer. Some of the features of dbPRO include:

- True dBASE III data file compatibility.
- True dBASE III program compatibility.
- Ability to access the **AMIGA**'s system routines, allowing the creation of mixed data/sound/graphics databases!
- Priced hundreds of $$ less than dBASE III.

**Special Introductory Price through May 1, 1988:** **$225.00**

**All orders after May 1, 1988:** **$299.95**

*For further information contact:*

**LampLighter Software, Inc.**

**3596 S. 300 W. #8**
**Salt Lake City, UT 84115**
**(801) 261-8177**

Coming soon: **AMIX**, a multiuser U*IX work alike for the **AMIGA**!

dealer inquiries invited

dBASE III is a trademark of Ashton-Tate, Inc.
dbPRO and dbPROFESSIONAL are trademarks of LampLighter Software, Inc.


**Step II**

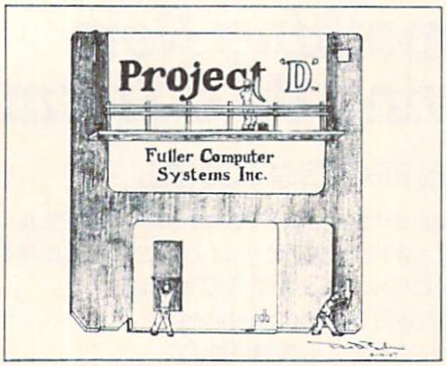
The object of the next step is to isolate the unknown column matrix on the left side of the equation. This is done by multiplying the equation by the inverse of the coefficient matrix. This leaves you with a two step reduction on the left side. The first reduction to recognize that the multiplication of a matrix by its inverse leave the unity matrix. The next reduction comes from the fact that any matrix multiplied by the unit matrix is that matrix (like multiplying something by 1). These operations leave you with the following:

$$[X] = [A]^{-1} * [K]$$

$$\begin{vmatrix} i1 \\ i2 \end{vmatrix} = \begin{vmatrix} 10 & -6 \\ -6 & 18 \end{vmatrix} * \begin{vmatrix} 24 \\ 0 \end{vmatrix}$$

*(continued)*

INTRODUCING.....



An Evolution in Disk Utilities for Amiga™ Personal Computers!

FEATURING

- An easy to use, friendly and intuitive user interface.
- A powerful and fast disk backup tool that lets you make backups of your copy-protected Amiga software.
- A disk editing tool that lets you edit raw MFM tracks, AmigaDOS sectors and AmigaDOS files (automatically calculating new checksums).
- A disk cataloging tool that lets you maintain lists of your personal, public domain and commercial software.
- A unique backup tool for duplicating other disk formats including MS-DOS/PC-DOS and Atari ST.
- An easy to read, informative user manual is included.
- This product is not copy-protected in any way.

NOW SHIPPING!

$49.95 EA.

Includes shipping and handling!
Arizona residents add 6.5% sales tax.

TO ORDER
Send check or money order to:
Fuller Computer Systems, Inc.
P.O. Box 9222
Mesa, Arizona 85204-0430
Or CALL (602) 835-5018

Amiga is a trademark of Commodore-Amiga, Inc.

Dealer Inquiries Invited

### Step III

To find the unknowns, simply multiply the inverse coefficient matrix by the resultant column matrix. This leaves you with a column matrix containing the values of the unknown variables.

Listed below is a set of BASIC program statements which, used with the matrix library routines, will produce the results described above.

```
'****** ' Test case for simultaneous solutions of loop '
equations for circuit 1 '****** ' Impedance Coefficient
matrix a(0,0)=10 a(0,1)=-6 a(1,0)=-6 a(1,1)=18

' Known resulting current matrix b(0,0)=24 b(1,0)=0
b(0,1)=0 b(1,1)=0


PRINT "Starting coefficient matrix " CALL pmats(2,a()) CALL
matinv(2,a()) PRINT "Inverse of coefficient matrix" CALL
pmats(2,a()) PRINT "Known result column matrix" CALL
pmats(2,b()) CALL matmul(2,2,1,a(),b(),c()) PRINT "Unknown
current matrix solution" CALL pmats(2,c()) PRINT "hit
return to continue" INPUT ch$
```

### Conclusion

These examples, along with the program segments, show how to get numeric solutions to the network topologies. The methods described here can be used perform time domain analysis, frequency domain analysis, and state-variable analysis. These analysis problems are performed with the aid of Fourier and Laplace transforms, which will be covered in future articles.

### Reference list

*Computer-Aided Analysis of Electrical Networks*, Thomas W. Davis, Ray W. Palmer

*Network Analysis*, M. E. Van Valkenburg

*Linear Control System Analysis and Design*, John J. D'azzo, Constantine H. Houpis

*Electric Networks*, Hugh Skilling

*Numerical Recipes: The Art of Scientific Computing*, William H. Press, Brain P. Flannery, Saul A. Terukolsy, William T. Vetterin

### Cramer's Rule Information Box

Cramer's rule is one way of solving simultaneous equations. This method uses the scalar results of determinants to solve the unknown variable. To observe the operations, begin with three equations and three unknown variables. It is given that the result of each equation has a known scalar result. This type of operation also assumes that the coefficients of the equations are known. The following is starting point to the solution.

$$\begin{aligned} a11^{**} + a12^{*}Y + a13^{*}Z &= A \\ a21^{**} + a22^{*}Y + a23^{*}Z &= B \\ a31^{**} + a32^{*}Y + a33^{*}Z &= C \end{aligned}$$

You should note that the equations are set up so that the same unknowns are in one column.

The determinant of the matrix, formed by the unknown column of the equations, will be used to find each of the unknown variables. This determinant scalar value will be referred to as D. It should look like this:

$$D = \begin{vmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{vmatrix}$$

If the scalar value is zero, you need not go further since there will be no solution to the set of equations (this would mean a division by a value of zero, which will cause most implementations of BASIC to stop execution of the program being run with an error message).

The solutions for X, Y, and Z are found by the following equations:

$$X = \frac{N1}{D} \qquad Y = \frac{N2}{D} \qquad Z = \frac{N3}{D}$$

The scalar values of N1, N2, and N3 are found by substituting the result column matrix (A, B, and C) in the column of the unknown variable to be solved. The next three matrices represent how to solve for the scalar N values.

$$N1 = \begin{vmatrix} A & a12 & a13 \\ B & a22 & a23 \\ C & a32 & a33 \end{vmatrix}$$

$$N2 = \begin{vmatrix} a11 & A & a13 \\ a21 & B & a23 \\ a31 & C & a33 \end{vmatrix}$$

$$N3 = \begin{vmatrix} a11 & a12 & A \\ a21 & a22 & B \\ a31 & a32 & C \end{vmatrix}$$

There are two special cases to this solution set. The first occurs if the result column matrix is all zeros. In this case, all the unknown values will be zero. The second case occurs if the scalar value of D is zero and the result column matrix is zero. This case may have a solution if the equations are set up with all but one in the form:

$$a11^{*} + a12Y = -a13Z$$
$$a21^{*} + a22Y = -a23Z$$

The solutions then can be found with reference to the Z variable by evaluation of the following matrices

$$X = \frac{\begin{vmatrix} -a13 & a12 \\ -a23 & a22 \end{vmatrix}}{\begin{vmatrix} a11 & a12 \\ a21 & a22 \end{vmatrix}} * Z$$

ATTENTION!
PROGRAMMERS & DEVELOPERS
The GS-1000
Graphics Library

- ★ A full featured INTERACTIVE Graphics Library maximizing the Amiga's™ powerful graphics capabilities!
- ★ One word commands eliminate hours and hours of programming time!
- ★ A complete library at your disposal, extensively documented and ready to use!
- ★ Built-in features include:

Device Independent Graphics • Multiple Coordinate Representations • Multiple Viewports and Screens • Zoom • Panning • Axial Scaling and Rotation • Rays • Palettes • Undo System w/Get Slate - Put Slate features • Color Mixing • Polymarkers • Brushes • Linestyles • Clipping • Interactive Text I/O • Debug Reporting facilities • Copy • Area Fills • Pattern Fills • PLUS standard features too extensive to list — over 90 commands in all •

PLUS A full featured Paint Program w/source to demonstrate the power of the Interactive Library.

Programmed in C, requires 512K memory.

"Amiga is a registered trademark of Commodore-Amiga, Inc."

SEND CHECK OR MONEY ORDER TO:
Rittinghouse Software Development Co.
P.O. Box 1272, Sioux Falls, SD 57101
VISA/MasterCard Customers Call:
Computer Business Center
(605) 342-6229 • Mon.-Fri. 9-5, Sat. 10-3
DEALERS INQUIRIES INVITED.

ONLY $49.95

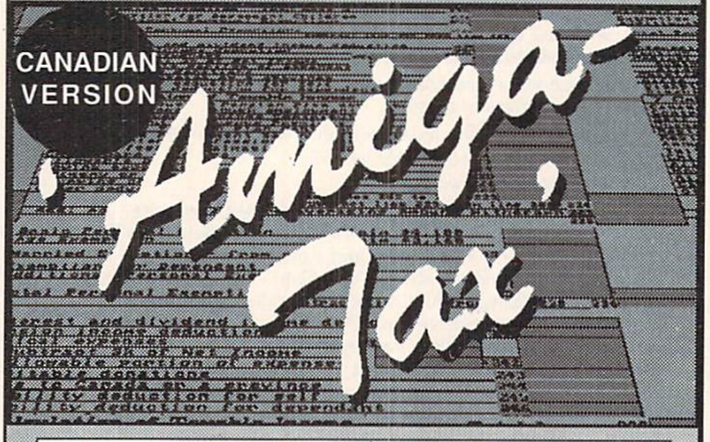
$$Y = \frac{\begin{vmatrix} a11 & -a13 \\ a21 & -a22 \end{vmatrix}}{\begin{vmatrix} a11 & a12 \\ a21 & a22 \end{vmatrix}} * Z$$

Again, no solution can be found if the evaluation of the determinant in the denominator is zero. It is therefore recommended that a check for a determinant operation returning a value of zero is made. This check can then be used to alter program flow to take the appropriate action.

```
' File:      matlib
' Function:  Floating point matrix library
' Language:  AmigaBASIC V1.2
' Date:      Aug 1987
' Author:    Robert W. Ellis
' Copyright 1987 Robert W. Ellis

' Change list:


' Global data allocation
DEFDBL a,b,c,h,r,s,nl,d
DIM a(10,10)
DIM b(10,10)
DIM c(10,10)
DEFINT i,n

' Subroutine shared variables
```

*(continued)*

Amiga-Tax II
Tax Planning and Preparation Software
Pay $69.95 once then receive
Yearly Updates for ONLY $29.95



CANADIAN VERSION
Amiga-Tax
-All Schedules, All provinces, Gov't Approved
-PRINTS tax returns No Manual transcribing
-Save time, money, and Aggravation
-Useful all year for tax planning
-Demo Disk available. Send $10
Datamax Research corp.
Box 5000, Bradford Ontario L3Z 2A6 416-775-2225

```
' subroutine mat_inv
DIM ipiv(20)
DIM  indxr(20),indxc(20)


' Main program and test area

PRINT "Electric Network solutions program"

'*****
' Test case for determinant method of solution
' to node equations for circuit 1
'*****
' Impedance Coefficient matrix
a(0,0)=10
a(0,1)=-6
a(1,0)=-6
a(1,1)=18

' N1 matrix formed by substitution
b(0,0)=24
b(1,0)=0
b(0,1)=-6
b(1,1)=18

' Solving for only one loop current, matrix
' corruption occurs
CALL  matdet(2,r1,b())
CALL  matdet(2,d,a())
r = r1/d
PRINT "Loop current i1 =";r
PRINT "hit return to continue"
INPUT ch$

'******
' Test case for simultaneous solutions of loop
' equations for circuit 1
'******
' Impedance Coefficient matrix
a(0,0)=10
a(0,1)=-6
a(1,0)=-6
a(1,1)=18

' Known resulting current matrix
b(0,0)=24
b(1,0)=0
b(0,1)=0
b(1,1)=0


PRINT "Starting coefficient matrix "
CALL  pmats(2,a())
CALL  matinv(2,a())
PRINT "Invers of coefficient matrix"
CALL  pmats(2,a())
PRINT "Known result colmun matrix"
CALL  pmats(2,b())
CALL  matmul(2,2,1,a(),b(),c())
PRINT "Unknown current matrix solution"
CALL  pmats(2,c())
PRINT "hit return to continue"
INPUT ch$

PRINT "end of main"
END
' end of main program

'************
' Subroutine :  mat_inv
' Function   :  invert a matrix
' Inputs     :
'************
SUB matinv(n%,a(2)) STATIC

SHARED ipiv()
SHARED indxr()
SHARED indxc()

PRINT "mat_inv"
n= n-1
FOR j=0 TO n
  ipiv(j) = 0
NEXT j
FOR i=0 TO n
  big=0!
  FOR j=0 TO n
    IF( ipiv(j) >< 1) THEN
      FOR k=0 TO n
        IF( ipiv(k) = 0 ) THEN
          big= ABS(a(j,k))
          irow = j
          icol = k
        ELSEIF( ipiv(k) > 1) THEN
          PRINT "singular matrix"
          RETURN
        END IF
      NEXT k
    END IF

  NEXT j

  ipiv(icol) = ipiv(icol)+1
  IF( irow >< icol ) THEN
    FOR l=0 TO n
      dum=a(irow,l)
      a(irow,l) = a(icol,l)
      a(icol,l) = dum
    NEXT l
  END IF

  indxr(i) = irow
  indxc(i) = icol

  IF( a(icol,icol ) = 0! )THEN
    PRINT "singular matrix"
    RETURN
  END IF

  pivinv = 1/a(icol,icol)
  a(icol,icol)=1

  FOR l=0 TO n
    a(icol,l) = a(icol,l) * pivinv
  NEXT l

  FOR ll=0 TO n
    IF( ll >< icol ) THEN
      dum = a(ll,icol)
      a(ll,icol) = 0
      FOR l=0 TO n
        a(ll,l) = a(ll,l) - a(icol,l) * dum
      NEXT l
    END IF
  NEXT ll
NEXT i

  FOR l= n TO 0 STEP -1
    IF( indxr(l) >< indxc(l)) THEN
      FOR k=0 TO n
        dum = a(k,indxr(l))
        a(k,indxr(l)) = a(k,indxc(l))
        a(k,indxc(l)) = dum
      NEXT k
    END IF
  NEXT l

END SUB
'******  end of subroutine mat_inv  ******

'*************
' Subroutine :  mat_mul
' Function   :  multiply two matrices
'
' Inputs     :  a  - input matrix A
'            :  ira- number of rows in A
'            :  irc- number of columns in A
'            :  b  - input matrix B
'            :  icb- number of columns in B
'            :  c  - output matrix C
'*************

SUB matmul(nra%,nca%,ncb%,a(2),b(2),c(2))  STATIC

nra = nra - 1
ncb = ncb - 1
nca = nca - 1

FOR i=0 TO nra
  FOR j=0 TO ncb
    c(i,j) = 0!
    FOR k=0 TO nca
      c(i,j) = c(i,j) + (a(i,k) * b(k,j))
    NEXT k
  NEXT j
NEXT i
END SUB

' end of subroutine mat_mul

'*************
' Subroutine :  mat_det
' Function   :  multiply two matrices
'
' Inputs     :  b     - input matrix B
'            :  nsize - size of square matrix
' Outputs    :  rdet  - determinant scalar
'*************

SUB matdet(nsize%,rdet,b(2)) STATIC
nsize = nsize -1
rdet = 1!
tol = 1E-10

FOR n=0 TO nsize
  ncol = n
  done = 0
  WHILE( done = 0 )
    pivot = b(n,n)
    IF( n = (nsize) ) THEN
      rdet = rdet * pivot
      done = 1

    ELSE
      IF( ABS( pivot ) > tol ) THEN
        FOR j=0 TO nsize
          b(n,j) = b(n,j) / pivot
        NEXT j
        k = n + 1

        FOR i=k TO nsize
          dmul = b(i,n)
            FOR j=n TO nsize
              temp = -dmul * b(n,j)
              b(i,j) = b(i,j) + temp
            NEXT j
        NEXT i

        rdet = rdet * pivot
        done = 1

     ELSE
       ncol = ncol + 1
       FOR j=0 TO nsize
         b(j,n) = b(j,n) + b(j,ncol)
       NEXT j
       IF( ncol > nsize ) THEN
         rdet = 0
       END IF

       IF( rdet = 0 ) THEN
         GOTO detend
       END IF
       done = 0
```

*(continued)*

Project Disk Special RAMTools WBExtras
Open
Close
Load
Unload
CLI
WBExecute
ViewText
SeePicture
LoadPicture
UnloadPicture

**WBEXTRAS**
*by Peter Dunlap*

Isn't it time you got the most from your Amiga?

Now, "WBExtras" is here and is specifically designed to enhance operation of the Amiga by the "New User" as well as the "Seasoned Programmer".

**For the Amiga User...**
Use of New Workbench Menus, "RAM Tools" and "WBExtras" provide access to ANY Workbench Tool from the Workbench Menu and allow "Multiple Icon Selection" without the use of the "SHIFT Key". Also, "New Execution Modes" permit a "Single Loading" of Workbench Tools for Multiple Task Execution. This results in "Optomized Memory Allocation" and "Reduced Disk Thrashing". For FULL System Memory, WBExtras will "PolitelyRetire" and RELEASE ALLOCATED MEMORY WITHOUT RE-BOOT. As a BONUS, several New Workbench Tools are included (See Menu).

**For the Amiga Programmer...**
WBExtras includes SOURCE CODE in "C" and "AmigaBASIC" for Workbench Tools using a NEW Programming Technique which provides "Optomized Memory Utilization". "Inter-Program Communication", and "Disk Access Queing".

Lynn's Luna
SYSTEM INNOVATIONS
P.O. Box 1308
Cañon City, CO 81212
303 275-5858

$39.95
Plus 3.00 for Shipping
Colorado Res. Add Sales Tax

**Amiga & AmigaBasic Workbench TM of Commodore - Amiga, Inc.**
***Dealer Inquiries Invited***

Even up the Score !!!

INVESTOR'S ADVANTAGE

Start using your Amiga to give you the Advantage in making better investment decisions! Color Graphics of Individual Stocks and General Market Trends. High Low Close, Moving Averages, Volume, Momentum and Relative Strength. Select the best performers with Relative Strength Rankings. Know when to get into and out of the market. Update stocks and/or mutual funds manually or automatically. Easy to use Communications included.

Only $99.95

See your local dealer or call:

Software Advantage Consulting Corporation
37346 Charter Oaks Blvd
Mt. Clemens, MI 48043 (313) 463-4995

Amiga and The Investor's Advantage are trademarks of their companies.

```
     END IF
   END IF
  WEND
NEXT n
detend:

END SUB
' ******  end of subroutine mat_det ******


'*************
' Subroutine :  mat_add
' Function   :  add two matrices
'
' Inputs     :  a   - Input matrix A
'            :  nra - Number of ROWs in matrix A
'            :  nca - Number of COLUMNs in matris A
'            :  b   - Input matrix B
' Outputs    :  c   - Output matrix C
'*************

SUB matadd(a(2),nra%,nca%,b(2),c(2))  STATIC
nra = nra - 1
nca = nca - 1

FOR i=0 TO nca
  FOR j=0 TO nra
    c(i,j) = a(i,j) + b(i,j)
  NEXT j
NEXT i

END SUB
'  ******  end of subroutine mat_add ******

'*************
' Subroutine :  mat_sub
' Function   :  subtract two matrices
'
' Inputs     :  a   - Input matrix A
'            :  nra - Number of ROWs in matrix A
'            :  nca - Number of COLUMNs in matris A
'            :  b   - Input matrix B
' Outputs    :  c   - Output matrix C
'*************

SUB matsub(a(2),nra%,nca%,b(2),c(2))  STATIC
nra = nra - 1
nca = nca - 1

FOR i=0 TO nca
  FOR j=0 TO nra
    c(i,j) = a(i,j) - b(i,j)
  NEXT j
NEXT i
END SUB
'  ******  end of subroutine mat_sub ******

'*************
' Subroutine :  mat_cpy
' Function   :  copy a matrix
'
' Inputs     :  a   - Input matrix A
'            :  nra - Number of ROWs in matrix A
'            :  nca - Number of COLUMNs in matris A
' Outputs    :  b   - Output matrix b
'*************
SUB matcpy(a(2),nra%,nca%,b(2))  STATIC
nra = nra - 1
nca = nca - 1

FOR i=0 TO nca
  FOR j=0 TO nra
    b(i,j) = a(i,j)
  NEXT j
NEXT i
END SUB
'  ******  end of subroutine mat_cpy ******

'*************
' Subroutine :  mat_scl
' Function   :  multipliy a matrix by a scalar
'
' Inputs     :  a   - Input matrix A
'            :  nra - Number of ROWs in matrix A
'            :  nca - Number of COLUMNs in matris A
'            :  scl - scalar value
' Outputs    :  a   - Output matrix A
'*************
SUB matscl(a(2),nra%,nca%,s!)  STATIC
nra = nra - 1
nca = nca - 1
PRINT "nra ";nra;" nca";nca;" scl";s!
FOR i=0 TO nca
  FOR j=0 TO nra
    a(i,j) = a(i,j) * s!
  NEXT j
NEXT i

END SUB
'  ******  end of subroutine mat_scl ******

'*********
'*  Start of support and debug routines
'*********
'*************
' Subroutine :  pmats
' Function   :  print a square matrix
'*************
SUB pmats(n%,a(2))  STATIC
n = n-1
FOR i=0 TO n
  PRINT " | " ;
  FOR j=0 TO n
    PRINT USING " ######.######";a(i,j);
  NEXT j
  PRINT " |"
NEXT i
END SUB
```

•AC•

# Benchmark Modula-2 Compiler

by Richie Bielak

"... You'll never want to program in Pascal again."

Until this fall, TDI Modula-2 was the only commercially available compiler for the Amiga. A public domain compiler was available, but its limited support of Amiga features and lack of documentation made it difficult to use.

I am always looking for better tools, so even though I had used TDI Modula-2 for over a year, I was very interested in the Benchmark Modula-2 compiler. I obtained a pre-release version in August, and I have been using it ever since.

### *What It Is*

The Benchmark Modula-2 package contains not only a compiler and a linker, but also an editor (with a built-in linker), many library modules, some neat utilities, and many demo programs. The package provides an entire Modula-2 development environment for the Amiga.

For the most part, the Benchmark compiler conforms to the language definition from the third edition of N. Wirth's *Programming in Modula-2*. It is implemented as a single-pass compiler. Since a single-pass compiler reads the source program only once, the compilation times are shorter. The benefit of faster compilation outweighs any disadvantages of single-pass compilation (such as having to declare objects before use). In fact, incredible compilation speed is one of best features of this product (see the benchmarks below).

Benchmark Modula-2 allows full access to all ROM Kernel, Intuition, and AmigaDOS routines. The interface to the Amiga is provided through library modules, in the usual Modula-2 fashion. Benchmark Modula-2 programs can be linked with Amiga binaries (for example, programs written in Amiga assembler), so you can develop interface modules for libraries other than the standard library.

The editor is based on the well-known Micro-Gnu-Emacs. As Gnu-Emacs is my favorite editor, I was very happy with the Benchmark editor. This editor has many nice features (as does Micro-Gnu-Emacs), including the ability to edit and view many files at once, ease of cutting and pasting between files, and creating keyboard macros.

The main disadvantage of an Emacs-like editor is the command syntax. All commands consist of somewhat cryptic control sequences, usually one or two characters. Once learned, however, the commands are fast and easy to use. To make learning easier, the Benchmark editor provides menus for the most common commands.

The major incentive for learning the Benchmark editor is the ability to invoke both the compiler and the linker directly from the editor with a single keystroke. This arrangement is similar to the Turbo Pascal environment. The compile-link-run cycle is discussed in detail below.

The library modules provided with Benchmark Modula-2 include the "standard" Modula-2 collection: InOut, Terminal, FileSystem, etc. and also included are all expected Amiga libraries: ROM Kernel, Intuition, and AmigaDOS. In addition, other groups of useful modules are supplied, including IFF support modules, "C-style I/O" modules, and "Simple Intuition" modules.

The "Simple" modules truly simplify access to the Amiga system. You can easily write programs that use many fancy Amiga features with these modules. The program in Listing #1 demonstrates use of the "SimpleWindows" and "SimpleGadgets" modules. The program opens a window, attaches a gadget to it, and waits for the user to click on the gadget. The size of the program is an indication of how "Simple" modules simplify programming.

In addition to the compiler, linker, and editor, the package also includes a number of useful programs. A program that reconfigures the compiler, a cross reference utility, a program that converts Amiga binaries into Benchmark Modula-2 object files, and a utility to quickly load all library and symbol files into memory are all included.

(continued)

Two utilities, M2STAT and ADDIMAGE, are somewhat unusual. M2STAT allows limited profiling of program execution. With this utility, you can see how many times each procedure in your program was called. This data can be extremely useful in speeding up Modula-2 programs. Furthermore, M2STAT can be used in testing and debugging—you can make sure all procedures were executed.

The ADDIMAGE program converts IFF files—brushes or pictures—into bitmaps that can be used directly in a program. ADDIMAGE appends this data to the executable file. Once your program is loaded, the image data is automatically placed in CHIP memory and can be used whenever images are needed. (Special modules allow easy access to all such images.) This arrangement eliminates the need for creation and initialization of large arrays containing bitmaps.

### *System Requirements*

Benchmark Modula-2 runs comfortably on a 512K Amiga with two drives. The compiler can also be used on a single-drive system, but some effort is required to squeeze all the necessary files onto one disk.

A special utility tailors the compiler to your particular system configuration. The M2CONFIG program modifies internal compiler parameters, thus controlling how much memory the compiler uses. Some parameters, such as heap size, code buffer size, identifier buffer size, and constant buffer size, can be adjusted. Memory can be saved by making the parameters small.

Once you reconfigure the compiler, you may run into a program too large to compile. For example, the compiler might run out of heap space, but this is only a minor problem. While in the Benchmark editor, you can temporarily modify the settings of the compiler parameters. If you have enough memory, your program should then compile without a problem.

I use Benchmark Modula-2 with two disk drives and 1.5 meg of RAM. All the libraries needed for compilation fit in the memory on my machine, with adequate room left for the editor and compiler. I use the Virtual Disk device from ASDG, so all the library files reside in VD0:. (By the way, the Virtual Disk software is included in the Benchmark distribution—it's a Shareware product.)

### *What You Get*

The Benchmark package consists of six disks and a two volume manual. Remember, this is the pre-release version, so the final version may include additional items. Two disks contain all the basic elements needed to set up the Modula-2 environment. The first disk contains the compiler, editor, linker, and extra utilities. The second disk is filled with library modules and a file which contains all the libraries compressed for quick loading into RAM. Three other disks contain demo programs and their sources, as well as sources to all IFF support modules.

The final disk contains Benchmark Modula-2 configured for a 512K system. If you boot the Amiga from this disk, you can immediately enter, compile, and run programs. You can start running Modula-2 programs without having to go through a painful installation process. Of course, if you have more than 512K of memory, you should eventually set up your system to take advantage of the extra RAM.

### *Documentation*

The documentation for Benchmark Modula-2, in its pre-release form, consists of two 400-page manuals. About 60 pages of pre-release notes are also included.

The first volume includes an installation guide, a tutorial for the editor, and separate chapters describing the compiler, linker, and editor in full detail. Further chapters document selected library modules. For each module, each exported procedure is described with examples.

The bulk of the second volume is one 350-page chapter of all library definition modules. An index of all identifiers exported by all modules is also included. For each exported object, the index shows the module where the object is declared, and the object's type. For example, the entry for the procedure "GetMsg" is this: "GetMsg Ports PROCEDURE."

The chapter which includes definition modules and the index are absolutely essential for all except the very trivial programs. You look here to find out how various procedures are declared, in order to determine the orders and types of parameters. Unfortunately, the index's format leaves much to be desired.

Ideally, the index should list the module, type, and page number on which the identifier is declared. Finding an identifier in a large module can be difficult. For example, the identifier "GadgetDown" is declared in the module "Intuition," but the "Intuition" definition module is 37 pages long. I hope the index is improved in the final release version.

The second volume also includes seven appendices. The first appendix contains the listing of all possible errors for the compiler, linker, or editor. Possible reasons for the error are given and a solution is suggested—a very nice touch. These explanations were very helpful. Other goodies in the appendices include a short glossary of related terms and a section on translating C programs to Modula-2.

I was also impressed by the release notes, which described the differences between the Beta and pre-release versions of the product. All changes were described in the release notes.

The corresponding documentation changes were also included. Finally, a four-page section gave hints on conserving memory, so the Benchmark product can be effectively used on a 512K machine.

Despite the few problems, the pre-release documentation for Benchmark Modula-2 was excellent. It was much better than the *final* documentation for some commercial products.

### *How It Works*

The compile-link-run cycle under Benchmark proceeds as follows. First, type the program into an editor buffer. Compile the current buffer (remember, there can be many buffers), by pressing the F2 function key. If the compilation fails, use the F1 function key to examine the errors. Each time you press F1, the cursor moves to the next error, and the error message is displayed at the bottom of the screen.

Once a clean compilation is achieved, the program is ready for linking. Before a program can be linked for the first time, the main module's name must be set. To do this, press F6 and type in the module's name. Linking begins when you press the F3 function key. Finally, press F4 to run the program.

Since you usually work on one program for a while, the sequence of events to compile, link and run the program is as follows: press F2 to compile, fix any errors, press F3 to link, and press F4 to run. You can now see how Benchmark program development is similar to Turbo Pascal environment.

Besides the obvious convenience of fewer keystrokes, the above development cycle has other advantages. First, the compiler is kept in memory, so it need not be loaded from disk for each compilation. The Benchmark linker is actually integrated into the editor (although a stand-alone version is provided), so the linker doesn't require loading either. Finally, the compiled program is read from the editor's buffer—that is, from memory. The compiler never reads the source from disk. This contributes to very fast compilation.

If you have enough memory to keep the library symbol and object files in memory, the compilations are lightning fast. On a 1.5 Mb Amiga, all libraries fit in memory. On a smaller machine, such as a 1 Mb Amiga 500, the most commonly used libraries can be placed in memory, and the compilations will still be very fast.

Placing the library files in many different directories presents no problem with Benchmark Modula-2. The compiler and linker allow specification of search directories. For example, the compiler first looks for a symbol file in the current directory; if the file is not there, it checks any directories named in the search path.

Programs produced by the Benchmark Modula-2 system can be run from CLI or Workbench without changes. The only exceptions are programs expecting command line arguments. Such programs must be "smart" and realize they have been started from Workbench.

### *Runtime Environment*

The runtime environment of the Benchmark Modula-2 program differs slightly from the environments of programs written in TDI Modula-2 or in C. For example, the Benchmark startup code opens all the ROM resident libraries by default. In this case, your program never needs to open Intuition library. This can sometimes be a disadvantage, because the programmer cannot check whether the correct release of Intuition is in use. This arrangement is very convenient for most programs, though.

A nice feature of Benchmark is that the support code takes care of the message sent to your program from Workbench (this applies only to programs started from Workbench). The startup code obtains the Workbench message's address and makes it available to the rest of the program, while the exit code returns the message to Workbench. In addition, a window for normal input and output is opened for programs started from Workbench.

By default, the generated code does not check for possible runtime errors. Separate compile options—embedded in the program as special comments—enable checking of array bounds, ranges, and overflow. Runtime checks do not help you recover from errors, though. To prevent the Amiga from crashing, you must install a handler to catch errors. Calling up the "InstallErrorHandler" procedure (from module "RunTimeErrors") at the start of the program accomplishes this installation. A complementary procedure, "RemoveErrorHandler," must also be called before termination.

If a runtime error occurs, the error handler takes control and displays a window which shows values in all machine registers, type of trap, and program counter. Unfortunately, these numbers are difficult to connect with the crash-causing line in your Modula-2 program.

The error window also lets you exit the program, or propagate the error to further handlers. The reason for propagating the error to other handlers is that Benchmark Modula-2 programs can be debugged with standard Amiga debuggers (such as Metadigm's Metascope). A compile and link option adds symbols to the Object and Executable files, and therefore allows symbolic debugging. I suspect the overall level of debugging is fairly low—you must look at machine instructions and addresses.

In TDI Modula-2, runtime errors are handled somewhat differently. Runtime errors are caught by the "Trapper" module. The information printed by "Trapper" after an error occurs allows you to determine which line caused the problem (it's not easy, though). In later releases of the TDI compiler, a Post-Mortem Debugger could be used to examine program variables and determine the problem line. On the other hand, TDI Modula-2 programs could not be debugged with symbolic debuggers.

### *Non-standard Features/Extensions*

Let's examine the contents of the pseudo-module SYSTEM. All expected types are exported from this module, including BYTE, WORD, LONGWORD, and ADDRESS. BYTE, WORD, and LONGWORD represent the basic data types of the 68000 processor. ADDRESS is defined as POINTER TO BYTE. These types are treated specially by the compiler. For example, ADDRESS is compatible with every pointer type, but arithmetic operations can be performed on ADDRESS variables.

SYSTEM also includes the usual TSIZE and ADR procedures. The TSIZE procedure returns the size, in bytes, of a given type. The ADR procedure obtains the address of an object. In most Modula-2 implementations, the ADR procedure works only with variables. The Benchmark compiler also allows use of ADR on string constants and procedures. This expression, for example, is legal: ADR ("Hello, world"). This extension is very convenient, since an Amiga programmer in the Amiga environment must often obtain addresses of strings.

Other procedures defined in SYSTEM include SHORT, LONG, SHIFT, SETREG, REG, and INLINE. SHORT and LONG perform type conversion between types whose lengths are two and four bytes. SHIFT procedure performs shifts (either left or right) by specified number of bits. Depending on whether the shifted variable is unsigned or signed, a logical or arithmetic shift is performed. The REG and SETREG procedures allow examination and modification of machine registers. Finally, INLINE procedure imbeds machine code (which must be in hex) in your Modula-2 programs.

The Modula-2 multi-tasking primitives—type PROCESS, and procedures TRANSFER and IOTRANSFER—are not implemented. These omissions can make transporting programs from other machines difficult; it does not, however, prevent you from writing multi-tasking programs on the Amiga. You can use the services of the ROM Kernel.

"Standard" procedures NEW and DISPOSE are also missing from the Benchmark implementation. These procedures were removed from the Modula-2 definition by Niklaus Wirth, so they weren't included here. You can use the ALLOCATE and DEALLOCATE procedures from the "Storage" module instead.

Since Benchmark Modula-2 is a one-pass compiler, all procedures must be declared before being called. This can be a problem when mutually recursive procedures are needed (i.e., procedure A calls B, and B calls A). To handle this situation, Benchmark Modula-2 allows FORWARD procedure declarations. In a FORWARD declaration, you write down only the procedure header (the procedure name and the parameters), rather than the actual code.

Two compile options allow you to depart from some standard Modula-2 features. Modula-2 is a case-sensitive language (so is C, by the way); that is, it distinguishes between upper- and lower-case characters. A compile option avoids this discrepancy.

Another compile option allows the "_" (underscore) character in identifiers. Again, "standard" Modula-2 identifiers are composed of letters or numbers. I discourage the use of such options; they make programs less portable to other systems.

### *Limitations and Problems*

The current version of the Benchmark compiler limits the sizes of code and variables that can exist in a single module. The maximum size of code generated for a single module is 32K . This is hardly a problem—the source must be very large to produce that much machine code. Anything the compiler can do to encourage small modules is good! Of course, a linked program, which consists of many modules, can be as large as necessary.

A more serious restriction is the size of variables that can be declared in a module. The limit is also 32K. It is the sum of all variables' sizes. This declaration, for example, is not possible:

```
VAR
    a1, a2, a3 : ARRAY [0..20000] OF CHAR;
```

There is a further limitation on the size of array subscripts—the subscript cannot be larger than 32766. Therefore, this declaration is not acceptable:

```
TYPE
      (* Subscript too large *)
      buff = ARRAY [0..40000] OF CHAR;
```

Most programs never come up against those restrictions. Therefore, few programmers need to worry about these limits.

Another size restriction is imposed on the size of set types. The largest set can include only 32 elements, meaning SET OF CHAR is not allowed. If a program needs larger sets, a special module, "LargeSets," is provided. You can construct sets with up to 65,536 elements using procedures from this module.

The current version of the Benchmark compiler does not contain a LONGREAL type. The REAL type is implemented using the Motorola FFP format. Consequently, the Amiga math library is used for the functions in the "MathLib0" module (sin, cos, log, etc.).

An "optimize" option was added to the linker in a later release of the TDI Modula-2. This option instructs the linker to remove the procedures never called from the final executable file. In many cases, this process results in a reduction of the linked program. Right now, the Benchmark linker does not have such an option. Surprisingly, the executable files produced by Benchmark are not that much larger than corresponding optimized TDI files (see benchmarks below).

***Benchmarks***

To evaluate the performance of the Benchmark Modula-2, I measured the compilation and link times, and the sizes of the object and executable files produced. I did not perform any execution time benchmarks; such tests did not seem necessary because I did not observe drastic execution speed differences between TDI and Benchmark Modula-2. Furthermore, for most users, these times are not important. Remember, much of the code a program executes comes from Amiga libraries, and the same libraries are used, regardless of the language.

I used three different programs in my benchmarks: a "Hello, World!" program, a "Simple Window" program, and a "Gadget Demo." The first program printed a single message; the second opened a window and wrote a message in it; the third opened a window, attached every type of gadget to it, and processed IDCMP messages.

In my first benchmark, I tried to attain maximum compilation and link speed by putting all sources and library files in memory. For Benchmark, I compiled and linked all programs from the editor. In the TDI benchmarks, I installed the compiler and the linker as resident, so there was no load time. The results of this "turbo-charged" benchmark are summarized below:

*** Benchmark Modula-2 ***

| | Compile time | Link time | Object Size | Exec Size |
|---|---|---|---|---|
| Hello | 1.0 | 1.0 | 148 | 2420 |
| Window | 4.2 | 2.0 | 391 | 1596 |
| Gadgets | 7.2 | 3.0 | 2470 | 8088 |

*** TDI Modula-2 (Version 3.00A) ***

| | Compile time | Link time | Object Size | Exec Size |
|---|---|---|---|---|
| Hello | 2.0 | 6.0 | 240 | 1548 |
| Window | 35.0 | 17.0 | 1376 | 1908 |
| Gadgets | 60.0 | 20 | 5240 | 5420 |

As you can see, the Benchmark Modula-2 compile times are amazingly fast. For the "Gadgets" demo, the Benchmark compile is almost nine times faster than TDI compiler. The Benchmark linker is even faster!

The code generated by Benchmark is also smaller than TDI's code. One reason the Benchmark compiler produced smaller objects modules is the generated code uses 16-bit relative addressing scheme whenever possible, so that only two bytes of address are needed to reference variables. The absolute (or 32-bit) addressing requires four bytes for every reference. Benchmark's code is also smaller because calls to "OpenLibrary" procedure are not needed; they are done instead by the startup code. Within TDI programs, all libraries must be opened explicitly.

Despite the fact that the Benchmark compiler produces more compact code, its executable files are larger. Benchmark's linker is not optimized to remove unused procedures. Note that the difference in size is not that significant, though—Benchmark executables are about twenty to thirty percent larger.

For a more diverse comparison, the same benchmarks were run with the Lattice C compiler. The results, summarized below, were collected by Brian Maguire, Software Coordinator of AMUSE (N.Y.C. user's group). The aim was to obtain the best possible times—therefore the compiler, linker (BLINK), libraries, and all sources were placed in memory.

*** Lattice C Version 3.10 ***

| | Compile time | Link time | Exec size |
|---|---|---|---|
| Hello | 4.0 | 16 | 10232 |
| Window | 22.0 | 16 | 6092 |
| Gadgets | 32.0 | 17 | 13068 |

My next benchmark determined how well both Modula-2 compilers run on a 512K system. For the Benchmark package, I used the boot disk preconfigured for a 512K machine. The setup still allowed the compiler to be kept in memory, but generated objects and library files remained on disk. In TDI's case, the compiler, linker, and all library files were kept on disk. For both compilers the T: directory was in RAM:. Here are the results of the test:

| | ***Benchmark*** | | *** TDI *** | |
|---|---|---|---|---|
| | Compile time | Link time | Compile time | Link time |
| Hello | 8.0 | 14.0 | 21.0 | 33.0 |
| Window | 18.0 | 20.0 | 106.0 | 150.0 |
| Gadgets | 23.0 | 23.0 | 147.0 | 209.0 |

Benchmark Modula-2 shines in this particular test. The total time to compile and link the "Gadget" demo under Benchmark is 43 seconds. The same program takes nearly 6 minutes to compile and link using TDI!

You can speed up TDI's linker by avoiding the "optimize" option. The resulting executables are very large, however, and the time saving is minimal (about 10 percent).

The final comparison I made was in the sizes of symbol files. A symbol file is generated when a DEFINITION MODULE is compiled. The symbol file is read when the compiler needs to look at imported identifiers. Smaller symbol files speed up compilations because the compiler has less data to read. Furthermore, smaller symbol files take up less space on the disks.

Listing #2 contains the DEFINITION MODULE I compiled to compare the sizes of symbol files. Here are the results:

```
** Symbol file size **


    ** Symbol file size **


     Benchmark |    TDI
      +-------------------+
    |  1869    |   11172   |
      +-------------------+
```

Again, you can see the dramatic difference in favor of the Benchmark package.

As you can see from the numbers, the performance of the Benchmark Modula-2 system is impressive!

*Summary*

I hope the information provided in this article helps you decide if Benchmark Modula-2 is for you. If you have been waiting for an Amiga version of Turbo Pascal, forget it! Get the Benchmark package and learn Modula-2. You'll never want to program in Pascal again. I think the Benchmark Modula-2 package is the best Amiga program development environment to date. I also received excellent support from Benchmark's author Leon Frenkel, both on the telephone and on the networks. None of my questions or suggestions went unanswered.

•AC•

## LISTING #1

```
(*
        Here are examples of "SIMPLE" modules
        supplied with the Benchmark Modula-2 package.

        Written by Richie Bielak 10/4/1987
*)
MODULE  Window;

FROM SYSTEM             IMPORT ADR;
FROM SimpleWindows IMPORT CreateWindow;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList,
                      FreeGadgetList, AddGadgetTextButton;
FROM Intuition        IMPORT WindowPtr, CloseWindow, IDCMPFlags,
                      IDCMPFlagsSet, WindowFlags, WindowFlagsSet,
                        GadgetPtr;
FROM Ports              IMPORT GetMsg, WaitPort, ReplyMsg,
MessagePtr;

CONST
   DefaultWindowFlags =
      WindowFlagsSet {Activate, WindowClose, WindowDrag,
                      WindowDepth, WindowSizing};
   DefaultIDCMPFlags =
      IDCMPFlagsSet {Closewindow, GadgetUp, GadgetDown};

VAR
   wp:   WindowPtr;
   msgp: MessagePtr;
   gp:   GadgetPtr;

PROCEDURE MakeGadget () : GadgetPtr;
  BEGIN
    BeginGadgetList ();
    AddGadgetTextButton (30, 20, ADR("Click here!"));
    RETURN EndGadgetList ();
  END MakeGadget;

BEGIN
  gp := MakeGadget();
  wp := CreateWindow (10, 20, 200, 50,
                      DefaultIDCMPFlags, DefaultWindowFlags,
                         gp, NIL, ADR ("Simple Window"));

  msgp := WaitPort (wp^.UserPort^);
  msgp := GetMsg (wp^.UserPort^);
  ReplyMsg (msgp);

  CloseWindow (wp^);
  FreeGadgetList (gp^)
END Window.
```

## LISTING #2

```
DEFINITION MODULE Dumbo;

  FROM Intuition IMPORT GadgetPtr;

  PROCEDURE Fly (gp : GadgetPtr);

END Dumbo.
```

*Avant-Garde Software*
*2213 Woodburn*
*Plano, TX 75075*
*214-964-0260*

# ROOMERS

by The Bandito

## *Inside AmiExpo . . . Deluxe Write R.I.P.?? . . . New 1.3 Printer Drivers . . . Kickstart 1.4*

The second AmiExpo was held in Los Angeles this January. With all the electronics on hand, and the AmiExpo humming with many choice bytes of information, the Bandito's sensors were at full gain. A few strategic wiretaps helped ... but there's no need to detail the Bandito's methods. Some show highlights:

As usual, Aegis showed some awe-inspiring new Allen Hastings tapes. Random bytes on the wire say that Allen is working on a full-length animated feature. Now that VideoScape offers more colors and features, Allen's animations will be even better. Beats the heck out of little models hung on wires, at any rate.

Also at AmiExpo, a new company named Innovision showed off their Video Effects 3D software. The software creates 3D title effects at 60 frames per second in high-res. The results are stunning—the titles look just like the professional effects you see on TV. With a retail price of $199.95, it's not for the casual hobbyist. Availability is the oft-promised "one month." Seems to the Bandito that most programmers work on a Mayan calendar, where the lunar period can be up to year long.

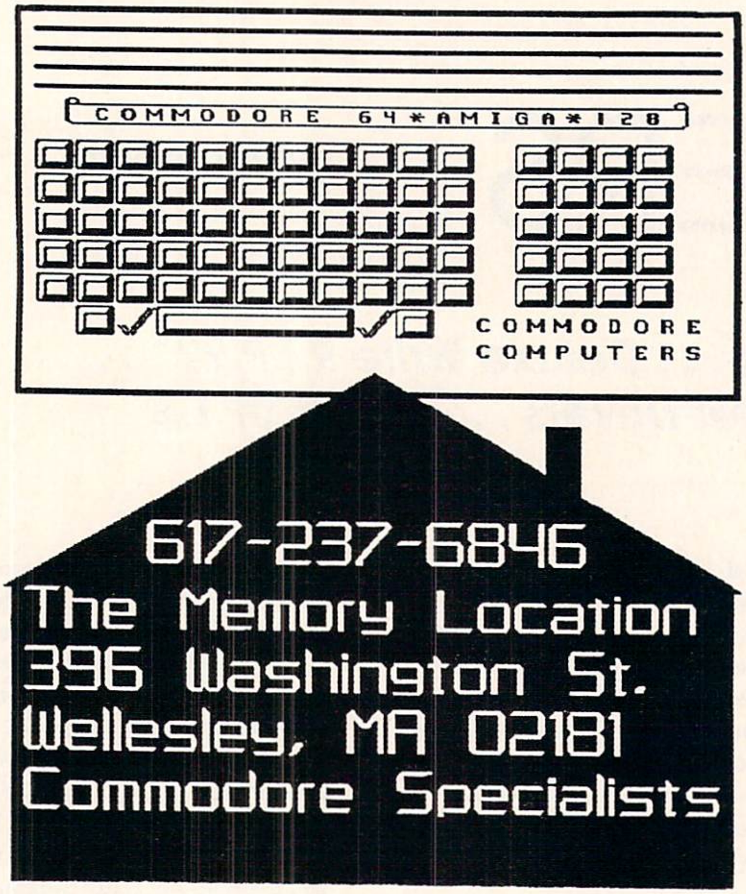
NewTek attracted big crowds with their Video Toaster, showing off their digital video special effects. Among the interested people were some engineers from ABC in New York who flew out especially to see the Toaster. Hmmm, could it be they're thinking of replacing some of that expensive hardware they now use for video effects?

ABC wasn't the only company interested in the Toaster. There are some very interested people at other hardware companies who would like to have that kind of product, especially since it outdates a few products that haven't even reached the market yet. Anybody know a good reverse engineer?

Well, well, the HAM paint program market is gonna get crowded—what with PhotonPaint from MicroIllusions and Deluxe Photo Lab (formerly D'Buddy from Digital Creations) from Electronic Arts challenging NewTek's Digi-Paint. At AmiExpo, NewTek announced Digi-Paint II, which will include texture mapping, anti-aliased fonts, work screens much larger than the monitor, and several new drawing modes, such as Colorize.

MicroIllusions showed PhotonPaint, but once they saw the Digi-Paint II spec sheet, they turned several interesting HAM colors and announced PhotonPaint II (before they even shipped PhotonPaint I!). After this, the joke at the show was, "What's PhotonPaint II gonna do? See Digi-Paint II's spec sheet!" Looks like the consumer will be the winner in this competition—all these programs have features so amazing that even the most jaded artist will find new ways to create digital paintings.

MicroIllusions also demonstrated Photon Video and Music-X. Photon Video, a new animation package, uses some talented Hollywood animators to create animation for the program, which makes for some exciting demos (the rabbit is particularly impressive). Music-X is an extremely powerful super sequencing program with a very clean interface. The only disappointment for some people is its inability to create SMUS files that Deluxe Music Construction Set or Sonix can read in. Maybe in a future version ... the Bandito advises you to let MicroIllusions know if you want this feature. The more noise you make, the faster you'll see it.



617-237-6846
The Memory Location
396 Washington St.
Wellesley, MA 02181
Commodore Specialists

No-show at AmiExpo: Mimetics. They told everyone they would attend, with Imagens and ReaSyns available, but they pulled out at the last minute. Seems like dat 'ole devil "production problems" is messing them. Mimetics is promising to meet the demand for Imagens "at any time." (Must be the same calendar programmers use.)

Everyone from Antic software was at the show, including the president and Tom Hudson. Are they considering moving their software to the Amiga? Could be, especially since Atari raised the price on the 520 ST by $100 in December, and the holiday sales fell through the floor.

Electronic Arts' word processor for the Amiga, Deluxe Write (as Batteries Included fans might remember it) is dead, the Bandito's sources tell him. EA's salespeople had been booking orders, but now they're cancelling. Looks like WordPerfect is the 800 pound gorilla that has crushed yet another competitor.

Commodore has seeded developers with new 1.3 printer drivers, and the drivers are getting rave reviews. The new drivers have better color correction, sideways printing that works properly, no lines in color printouts (a problem on printers like the JX-80) and much faster printing (no more nasty delays for each line).

By the way, the Bandito has heard about a no-holds-barred battle in the marketplace between the Xerox 4020 and the HP PaintJet color printers. The Bandito predicts the red ink in the street will be from Xerox. The prices of the two printers are the same and the Xerox has better color saturation, but the PaintJet is faster, quieter, smaller, and much more efficiently marketed. Look at the way the LaserJet has dominated the laser printer market. Xerox, a word to the wise: give up on the color printer and try to get a good, cheap, color copier out there before Sharp eats your lunch in that market.

Yes, Virginia, there is a KickStart 1.4 in progress. Just when you thought it was safe to burn 1.3 into ROM, it looks like Commodore's taking a peek at the Macintosh II, 'cause 1.4 will offer support for higher resolution screens and even multiple monitors! Better start saving now—those big-screen color monitors are expensive.

The Amiga is being featured on *Computer Chronicles*, the half-hour computer show on PBS (shown all over the US and Canada). Products covered include MicroIllusions' Music-X and Photon Video, and NewTek's Digi-View, Digi-Paint, and Video Toaster.

Finally, Commodore is rumored to be striking a deal with Toys 'R' Us to get some of Commodore's new low-cost IBM clones on the shelves, since Nintendo has beaten up the market for the C64. C64 hardware sales are dropping, so the A500 looks to be the savior, if Commodore can make it as popular an item as the C64 has been. It's going to take some time, but Commodore expects that the next holiday season will be very, very good for the Amiga 500. Let's help out by convincing everyone to buy an A500 instead of a Nintendo.

•AC•

# the BIG Picture

*by Warren Ring*

## 68000 Assembler Language Programming
## The Unified Field Theory Discovered!

Ever since I started using assembler languages, I have faced an obstacle that makes assembler really difficult to work with: Arguments to the various system calls are of all shapes and sizes. Disk I/O requires one convention of arguments. Character I/O requires another. Certain functions, such as binary-to-ASCII conversion and floating-point packages, simply don't exist (except in shops having the manpower to invent them).

Just as CP/M was becoming obsolete, I began to get a handle on a structure of argument passing that would be universal: a unified field. The requirements for a unified field are tough. It must handle text. It must handle binary data. It must be of different lengths, defined in different places in the program. There can be no artificial limits (like 64K) to the field. It must have hooks, so routines that manipulate the field can detect the field's length and avoid clobbering data following the field. And there must be hooks for fast manipulation—no linear scanning to find the end.

You can use the field to pick off text arguments on the command line. You can pass it directly to the routine to open a disk file. You can convert it to a binary integer. You can copy it, concatenate onto it, extract from it. Anything! This discovery is the theme of my next few columns.

### String Buffers Revealed

The key to making these high-level routines interface with one another is a common field introduced as the "string buffer." Buffers that hold strings are nothing new, but the unified field solves all the problems I have ever encountered in all kinds of arguments. String buffers solve problems your C friends tangle with every day. In my opinion, the two main credentials of a high-level language are rudimentary protection of code and data (subscript boundary checking and argument type checking), and a high number of functions debugged per hour. (C seems to lack the former, but that's another article.)

String buffers consist of three fields, shown in the following example:

```
Buffer1Length           equ 10
Buffer1
            DC.L        Buffer1Length,0
            DS.B        Buffer1Length
            CNOP        0,2
```

The first field is a long integer specifying the physical size of the data field in bytes (the maximum length of a string you wish to hold). Routines that write into the string buffer must know the size of the data field to avoid writing beyond that string buffer and destroying unrelated data or stack variables.

The second field is a long integer specifying the current usage of the data field in bytes (the length of the string in the buffer).

The third field is the data field itself. In the example above, the current usage is zero, and the data field is uninitialized. The CNOP statement is an assembler directive that aligns the program counter to the next word boundary. If Buffer1Length is an odd number, the assembler throws in an extra byte at the end of the buffer, so whatever follows is word-aligned. Another example follows:

```
GraphicsLib
            DC.L        12,12
            DC.B        'graphics.lib'
            CNOP        0,2
```

Use this example for a CONSTANT string—one that doesn't change. The string buffer has a maximum capacity of 12 bytes. It is currently using all 12 bytes, and "graphics.lib" is the data. Notice that the string has no terminating zero byte.

The functions I have developed involve both macros and subroutines. The subroutines require the arguments to be in specified registers; the associated macros perform that setup. Table 1 lists and describes these functions. There are too many functions to describe in only one column, so this topic will be continued in future articles. In previous articles, I showed examples of opening files, reading bytes, and sending them out to the CLI window. This macro and library package makes those functions far more uniform and much easier to use.

*(continued)*

*Table 1*

## Function Summary

*These functions handle disk, console, and port I/O:*

| | |
|---|---|
| **Open** | Opens a disk file |
| **Create** | Creates a disk file |
| **Delete** | Deletes a disk file |
| **Read** | Reads from a disk file, console, or port |
| **Write** | Writes to a disk file, console, or port |
| **WritCon** | Writes to the console |
| **Seek** | Seeks a specified position in a disk file |
| **Close** | Closes a disk file |

*These functions handle literal output to the console:*

| | |
|---|---|
| **Display** | Displays an immediate text string |
| **Crlf** | Displays a CR/LF |
| **Space** | Displays a space character |

*These functions perform string handling and are best described by BASIC or C language equivalent expressions:*

| | |
|---|---|
| **Left** | A$=LEFT$(B$,I), gets leftmost chars |
| **Mid** | A$=MID$(A$,I,J), gets middle chars |
| **Right** | A$=RIGHT$(A$,I), gets rightmost chars |
| **StrCpy** | A$=B$, copies one string to another |
| **StrCat** | A$=A$+B$, appends one string to another |
| **StrLen** | I=LEN(A$), gets a string length |
| **ItoA** | I =VAL(A$), converts integer to ASCII |
| **AtoI** | A$=STR$(I), converts ASCII to integer |
| **IsHexASCII** | Indicates whether or not a string is composed of hex-ASCII char's ("0" -"9", "A" - "F") |
| **AtoX** | I=HEX$(A$), converts a string of hex-ASCII chars to an integer |
| **X8toA** | A$=HEX8$(I), converts a 4-byte integer to a string of 8 hex-ASCII chars |
| **X4toA** | A$=HEX4$(I), converts a 2-byte integer to a string of 4 hex-ASCII chars |
| **X2toA** | A$=HEX2$(I), converts a 2-byte integer to a string of 2 hex-ASCII chars |
| **X1toA** | A$=HEX1$(I), converts a 4-bit integer to a string of 1 hex-ASCII char |

*These functions handle scanning of a string buffer to pick up words (including punctuation), alphanumeric words, or single characters:*

| | |
|---|---|
| **SetScan** | Sets scanning to the beginning of a string buffer |
| **Scanw** | Gets the next text word |
| **Scana** | Gets the next alphanumeric word |
| **Scanc** | Gets the next character |

*These functions handle housekeeping at the beginning and end of a program:*

| | |
|---|---|
| **Start** | Performs housekeeping at the beginning of a program; sets scanning to the command line residue |
| **Exit** | Performs housekeeping at the end of a program, and exits |

*This function creates an empty string buffer:*

| | |
|---|---|
| **StrBuf** | Directs the assembler to define an uninitialized string buffer |

At this point, the program package is not a library. It is an included file at assembly time. As soon as I discover how to make a library file from which the linker can selectively pull modules, I'll show you how it's done.

### The Example Program

The macros and routines that incorporate string buffers are shown in Table 1. This month I will discuss a small test program that demonstrates some of them. Additional details regarding the actual arguments for functions used this month are shown in Table 2.

This test program consists of 22 instruction statements. You run it by entering the name of the program, "test," on the CLI line, followed by any number of words, separated by any number of spaces. This program picks up the first six English words on the CLI line following "test," and, for each word, performs the following steps: 1) displays that word; 2) converts that word, if possible, to a 4–byte signed integer; 3) adds that integer to a total; 4) converts that integer back to an ASCII string; 5) displays that string. A concatenation of the six words is displayed on the next line, and the total of the signed integers is displayed on the next line. All that in 22 instruction statements! Who says you need a high-level language for fast development?

In case you missed last month's column, you may assemble the test file by entering:

```
assem test.asm -o test.o -l test.lst
```

This produces object file "test.o" and list file "test.lst." Assuming you have no errors, you copy the "amiga.lib" file from the Metacomco disk to the directory you have assigned to libs:, and enter:

```
alink test.o to test library libs:amiga.lib
```

Listing 1 shows the source code for our test program. This program exercises the functions described this month. Assemble the source file using the Metacomco assembler. Remember, macros and library labels are case-sensitive.

In the example program, arguments that reference string buffers are generally IMMEDIATE, shown with a number sign before the label. Arguments that are integers are generally DIRECT, shown simply as labels. There may be exceptions, depending on what you want to do.

Listing 2 shows the "macros.asm" file. Listing 3 shows the "warlib.asm" file. Listing 4 shows a sample run of TEST.

Macro arguments are separated by commas. There cannot be spaces between or within arguments, unless the argument is surrounded by angle brackets (<>). The Display macro requires angle brackets if there are any spaces in the text.

The comments in the listings are an implementation of what I described two months ago in this column. Each is a complete, machine-independent, English sentence. The only machine-dependent comments are in parentheses. You can follow the exact flow of logic without knowing any 68000 assembler instructions.

### Debuggers

I neglected to mention one tool in the last issue—a debugging program. A debugging program is as essential to an assembler programmer as a hammer is to a carpenter. The only debugger I have found is MetaScope by Metadigm, Inc. I worked with MetaScope for a few hours and found it quite good. You can create as many windows as you wish, each displaying a disassembly, a section of memory, register values, and more. It's very powerful.

There are a few inconveniences, however. First, you must read the entire 35-page instruction manual before you can really get a handle on the program. (You have to invest a couple of hours just to learn MetaScope, but that isn't so bad.) Second, once your program exits, MetaScope does not let you execute it again without unloading it and reloading it.

Third, when reloading, the program loads into a different location in RAM. This means all your absolute addresses change. It also seems to indicate that some resource is not getting freed during the unload operation. After exiting MetaScope and entering Uedit, I have had visits from the Guru (indicating memory corruption), but I have not ascertained that MetaScope was at fault. Overall, I find MetaScope to be everything it's advertised to be.

*Table 2*

## This Month's Function Detail

*This table shows the argument values passed to each macro used in this month's example program.*

| Name | # Args | |
|---|---|---|
| **Start** | 0 | |
| **Exit** | 0 | |
| **WritCon** | 1 | SB containing text to send to the console |
| **SetScan** | 1 | SB to scan later |
| **Scanw** | 1 | SB to receive the next word scanned |
| **StrCat** | 2 | First arg: SB to concatenate<br>Second arg: SB to be concatenated onto |
| **AtoI** | 2 | First arg: SB to be converted<br>Second arg: integer result |
| **ItoA** | 2 | First arg: integer to be converted<br>Second arg: SB result |
| **Crlf** | 0 | |
| **Space** | 0 | |
| **Display** | 1 | SB to be displayed on the console |
| **StrBuf** | 2 | First arg: a string buffer label<br>Second arg: the maximum length of the string buffer |

SB = *the location of a string buffer*

### Parting Words

If you don't want to type in the listings in this article, I may be able to enter them into a bulletin board system. I regularly visit the Los Angeles Amiga User's Group BBS (LAAUG Line) at (213) 559-7367, and the "1939" BBS at (818) 368-4248. You may leave messages for me at either of these boards. There is one problem—because of publishing deadlines, files I leave on a BBS are likely to be two months ahead of what appears in this column.

*(continued)*

# Blitzkrieg at the Ardennes

(Made for the Amiga™ WW II battle simulator)

Early in December 1944 America was waiting the return of its victorious armies in Europe. But on the other side of the Seigfried Line, Hitler had other plans, launching his last major surprise attack of the war, and introducing the new 80-ton Tiger tank. You are in command of either the Allied or the German forces. The die is cast. Make your **command** decision. This is the game that had to wait for the Amiga™.

**FEATURES:**

- **Easy to understand rules and concepts**
- **Detailed full-color graphics**
- **Realistic sound effects**
- **One or two players**
- **Seigfried Line**
- **Artillery**
- **Supply**
- **Terrain effects**
- **V-2 Rocket attacks**
- **German saboteurs**
- **Paratroopers**
- **Play customization**
- **Save Game function**
- **Handicapping system for play balance**
- **Weather**
- **Fuel Dumps**
- **Aerial bombardment**

**$52.95**
plus $3.00
post. & hand.
Visa & MasterCard
Check, Money Order
or C.O.D.
Mass. Res. 8% tax

COMMAND Simulations
Building 3, 297 North Street
Hyannis, Mass. 02601
**(617) 790-1994**
Requires 512 K Amiga™ or Amiga 500™

**ARexx ... *The REXX Language for the Amiga***

ARexx is a multitasking implementation of the REXX language, an elegant high-level language especially suited for macro-processing and general programming tasks. Its clean, simple syntax makes REXX easy to learn ... an ideal "first language." And the powerful language features will appeal to experienced programmers as well!

- Interactive, interpreted operation
- Exceptional string-handling facilities
- Built-In library with over 75 functions
- Built-In source-level debugger
- Compact code — only 32K!

ARexx defines a command interface that allows it to communicate with other software. The list of Amiga software products that support this interface includes:

- TxEd-Plus from C. Heath/MicroSmiths
- C.A.P.E. 68K Assembler from Inovatronics
- AmigaTeX V2.9A from Radical Eye Software

Written by the author of "ConMan," ARexx sells for ...

**Only $49.95!**

Send check or money order for $49.95 plus $2.00 shipping to:
(MA orders please add 5% sales tax)

William S. Hawes
P.O. Box 308
Maynard, MA 01754
(617) 568-8695

Amiga is a trademark of Commodore-Amiga, Inc.

## *Listing 1*
## *TEST.ASM Source Code*

```
; TEST.ASM   This program picks up the first six
; English words on the CLI line following "TEST",
; and, for each word, displays that word, converts
; that word, if possible, to a 4-byte signed
; integer, adds that integer to a total, converts
; that integer back to an ASCII string, and
; displays that string.  A concatenation of the
; six words is displayed on the next line, and the
; total is displayed on the next line.

   section code

   include "libs:types.i"
   include "libs:dos.i"
   include "macros.asm"

   Start                :Perform startup
                        ; housekeeping
   MOVE.L  #5,D2        ;Set the counter to 5
X1
   Scanw   #Word        ;Display the first (next)
                        ; english word from
   Display '-'          ; the CLI residue,
                        ; surrounded by "-"
   WritCon #Word
   Display '-'
   Space                ;Display " "
   AtoI    #Word,Value  ;Convert the word to an
                        ; integer, and back to a
   ItoA    Value,#Word  ; string
   WritCon #Word        ;Display the resulting
                        ; string
   StrCat  #Word,#Buffer;Concatenate the word onto
                        ; the final string
   Crlf                 ;Display a CR
   MOVE.L  Value,D0     ;Add the integer value to
   ADD.L   D0,Total     ; the total
   DBRA    D2,X1        ;Decrement the counter
                        ;If the counter is not yet
                        ; negative, then jump to X1
   WritCon #Buffer      ;Display the final string
   Crlf                 ;Display a CR
   Display <'The total is: '>
   ItoA    Total,#Buffer:Convert the total integer
   WritCon #Buffer      ; to a string, and display it

   Crlf                 ;Display a CR
   Exit                 ;Perform ending
                        ; housekeeping, and exit

   include "warlib.asm"

   section data

   StrBuf  Buffer,80
   StrBuf  Word,16

Value  DS.L    1
Total  DC.L    0

   end
```

## *Listing 2*
## *The Macro File "macros.asm"*

```
; MACROS.ASM   Macros by Warren A. Ring
;
; These macros place the passed arguments into
; the correct registers for the library routines

   XREF     _LVOOpenLibrary
   XREF     _LVOOpen
   XREF     _LVOClose
   XREF     _LVOInput
   XREF     _LVOOutput
   XREF     _LVORead
   XREF     _LVOWrite
   XREF     _LVOCloseLibrary

   XREF     _AbsExecBase

Start       MACRO
            MOVE.L  A7,SystemSP
            JSR     OpenDOSLibrary
            ENDM

Exit        MACRO
            JSR     CloseDOSLibrary
            MOVEA.L SystemSP,A7
            RTS
            ENDM

WritCon     MACRO
            MOVEM.L D0/A0-A1,-(A7)
            MOVE.L  ConOut,A0
            MOVE.L  \1,A1
            JSR     WriteFile
            MOVEM.L (A7)+,D0/A0-A1
            ENDM

Write       MACRO
            MOVEM.L A0-A1,-(A7)
            MOVE.L  \2,A0
            MOVE.L  \1,A1
            JSR     WriteFile
            MOVEM.L (A7)+,A0-A1
            ENDM

SetScan     MACRO
            MOVEM.L A0,-(A7)
            MOVEA.L \1,A0
            ADDQ.L  #4,A0
            MOVE.L  (A0),ScanCounter
            ADDQ.L  #4,A0
            MOVE.L  A0,ScanPointer
            MOVEM.L (A7)+,A0
            ENDM

Scanw       MACRO
            MOVEM.L A0,-(A7)
            MOVEA.L \1,A0
            JSR     Scanw_
            MOVEM.L (A7)+,A0
            ENDM
```

```
StrCat      MACRO
            MOVEM.L A0/A1,-(A7)
            MOVEA.L \1,A0
            MOVEA.L \2,A1
            JSR     StrCat_
            MOVEM.L (A7)+,A0/A1
            ENDM

AtoI        MACRO
            MOVEM.L D0/A0,-(A7)
            MOVE.L  \1,A0
            JSR     AtoI_
            MOVE.L  D0,\2
            MOVEM.L (A7)+,D0/A0
            ENDM

ItoA        MACRO
            MOVEM.L D0/A0,-(A7)
            MOVE.L  \1,D0
            MOVE.L  \2,A0
            JSR     ItoA_
            MOVEM.L (A7)+,D0/A0
            ENDM

Crlf        MACRO
            JSR     DisplayCrlf
            ENDM

Space       MACRO
            JSR     DisplaySpace
            ENDM

Display     MACRO
            MOVEM.L D0,-(A7)
            Write   #Displ\@,ConOut
            BRA     Displ_3\@
Displ\@     DC.L    Displ_2\@-Displ_1\@
            DC.L    Displ_2\@-Displ_1\@
Displ_1\@   DC.B    \1
Displ_2\@
            CNOP    0,2
Displ_3\@   MOVEM.L (A7)+,D0
            ENDM

StrBuf      MACRO
            CNOP    0,2
\1
            DC.L    \2,0
            DS.B    \2
            CNOP    0,2
            ENDM
```

## *Listing 3*

## *The Library File "warlib.asm"*

```
; WARLIB.ASM   Library routines by Warren A. Ring

OpenDOSLibrary

;This routine opens the DOS liibrary

   MOVEM.L D0/A1/A5-A6,-(A7);(Push registers)

   MOVE.L  A0,ScanPointer  ;Set the ScanPointer
   MOVE.L  D0,ScanCounter  ; and ScanCounter to
                           ; scan the CLI line

   MOVE.L  _AbsExecBase,A6 ;Open the dos library
   LEA.L   DosName,A1
   MOVEQ   #0,D0           ;(Specify any version)
   JSR     _LVOOpenLibrary(A6)
   MOVE.L  D0,DosLibraryHandle

   MOVE.L  DosLibraryHandle,A5;Save the console
   JSR     _LVOInput(A5)   ; input handle locally
   MOVE.L  D0,ConIn

   MOVE.L  DosLibraryHandle,A5;Save the console
   JSR     _LVOOutput(A5)  ; output handle locally
   MOVE.L  D0,ConOut

   MOVEM.L (A7)+,D0/A1/A5-A6;(Pop registers)

   RTS                     ;Return

DosName  DC.B    'dos.library',0
   CNOP    0,2
```

**NEW PRODUCTS** FOR THE **AMIGA 500**

**Announcing the PB-512 Memory Board**

Low cost 512K plug in memory board with clock/calender

Available in 0K configuration,
(User installs 16 - 256K RAM chips)
Also available with 512K installed.
Both boards utilize the same clock chip as the Commodore memory board, but use a **longlife trouble free lithium battery.**

NOW IN STOCK 0K $ 89.95
512K $149.95

**Announcing the AMIGA 500 Monitor Stand**

Quality Real Oak construction, with a light oak finish Designed for the Amiga 500, with custom cutouts for easy disk drive access, with the keyboard either pulled out for use, or retracted for game play.

NOW SHIPPING $ 44.95

Look for lowcost 1.5MB internal memory boards for the Amiga 500, and the Amiga 1000 in the 2nd quarter of 1988.

DEALER INQUIRIES WELCOME

To order contact
**CPE SYSTEMS**
6901 E. Earll Dr.
Scottsdale, AZ 85251
or call (602) 949-8805
M-F 10am-6pm, Sat 10am-2pm Mtn. time
Amiga is a trademark of Commodore-Amiga Inc.

```
CloseDOSLibrary

;This routine closes the DOS library

   MOVEM.L A1/A6,-(A7)     ;(Push registers)

   MOVE.L  _AbsExecBase,A6 ;Close the dos library
   MOVE.L  DosLibraryHandle,A1
   JSR     _LVOCloseLibrary(A6)

   MOVEM.L (A7)+,A1/A6     ;(Pop registers)

   RTS                     ;Return


ReadFile

;This routine reads a record from a file

;In:  A0 -> the file handle
;     A1 -> a string buffer to receive the data from the file
;Out: D0 - the number of characters read from the file
;
;Note: This routine overwrites any data already in the string buffer

   MOVEM.L D1-D3/A0-A1/A5,-(A7);(Push registers)

   MOVE.L  A0,D1           ;(File specifier)
   MOVE.L  A1,D2           ;(Buffer location)
   ADDQ.L  #8,D2
   MOVE.L  (A1),D3         ;(# of bytes to get)
   MOVEM.L A1,-(A7)
   MOVE.L  DosLibraryHandle,A5
   JSR     _LVORead(A5)
   MOVEM.L (A7)+,A1
   ADDQ.L  #4,A1
   MOVE.L  D0,(A1)

   MOVEM.L (A7)+,D1-D3/A0-A1/A5;(Pop registers)

   RTS                     ;Return
```

*(continued)*

```
WriteFile

;This routine writes a record to a file

;In:  A0 -> the file handle
;     A1 -> a string buffer containing the data for the file
;Out: D0 - the number of characters written to the file
;          (-1 - write error, possibly disk full)

   MOVEM.L D1-D3/A0-A1/A5,-(A7);(Push registers)

   MOVE.L  A0,D1             ;(File specifier)
   MOVE.L  A1,D2             ;(Buffer location)
   ADDQ.L  #8,D2
   ADDQ.L  #4,A1
   MOVE.L  (A1),D3           ;(# of bytes to write)
   MOVE.L  DosLibraryHandle,A5
   JSR     _LVOWrite(A5)

   MOVEM.L (A7)+,D1-D3/A0-A1/A5;(Pop registers)

   RTS                       ;Return


DisplayCrlf
          WritCon #DisplayCrlf_7
          RTS
DisplayCrlf_7
          DC.L     1,1
          DC.B     10
          CNOP     0,2

DisplaySpace
          WritCon #DisplaySpace_7
          RTS
DisplaySpace_7
          DC.L     1,1
          DC.B     ' '
          CNOP     0,2


StrCat_            ;BASIC Fn: B$ = B$ + A$

                   ;In: A0 -> source buffer (A$)
                   ;    A1 -> dest buffer (B$)

   MOVEM.L D0-D2/A0-A1,-(A7) ;(Push registers)

   MOVE.L  (A1),D0 ;(Set D0 to the max length of
                   ; B$)
   ADDQ.L  #4,A1   ;(Make A1 point to the current
                   ; usage of B$)
   ADDQ.L  #4,A0   ;(Make A0 point to the current
                   ; usage of A$)
   MOVE.L  (A1),D2 ;(Set D2 to the current usage
                   ; of B$)
   MOVE.L  (A0),D1 ;Set the byte counter (D1) to
                   ; the current usage of A$
   ADD.L   (A1),D1 ; plus the current usage of B$
   CMP.L   D0,D1   ;If the byte counter > the max
                   ; length of B$,
   BLE     Strcat_1; then set the byte counter to
                   ; the max usage of B$
   MOVE.L  D0,D1
Strcat_1
   MOVE.L  D1,(A1) ;Set the current usage of B$
                   ; to the byte counter
   SUB.L   D2,D1   ;Subtract the current usage of
                   ; B$ from the byte counter
   TST.L   D1      ;If the byte counter is now
   BEQ     Strcat_9; zero, then jump to Strcat_9
   ADDQ.L  #4,A0   ;(Make A0 point to the first
                   ; data byte of A$)
   ADDQ.L  #4,A1   ;(Make A1 point to the byte
                   ; after the last byte currently
   ADD.L   D2,A1   ; used by B$)
   SUBQ.L  #1,D1   ;Copy the bytes from A$ to B$
Strcat_2
   MOVE.B  (A0)+,(A1)+
   DBRA    D1,Strcat_2
Strcat_9

   MOVEM.L (A7)+,D0-D2/A0-A1;(Pop registers)

   RTS             ;Return


Scanw_

;This routine scans a string for the next visible
; word
```

COMPUTER MART

Your Texas Amiga Source
Immediate Access to over 400 Amiga Titles.
Prices too low to print!

We stock Amiga Software and Peripherals
For A500, A1000, & A2000.
Mon. thru Fri. 10:00 AM-7:00 PM, Sat. 12:00-5:00 PM

CALL TOLL FREE
800-443-8236

CUSTOMER SERVICE
409-560-2826

Computer Mart • 105 Lynn Street • Nacogdoches, TX 75961

```
;In:  ScanPointer => the next byte in the string
;                     to be scanned
;     ScanCounter = the number of bytes in the
;                    string remaining to be
;                    scanned
;     A0 => the dest buffer
;Out: ScanPointer, ScanCounter are updated
;
;Note: This routine skips over spaces, tabs, and
; other invisible characters

   MOVEM.L D0-D2/A0-A2,-(A7)    ;(Push registers)

   MOVE.L   (A0)+,D1;(Set D1 to the max length of
                     ; A$)
   MOVE.L   A0,A1    ;(Make A1 point to the current
                     ; length of A$)
   MOVE.L   #0,(A1) ;Set the current usage of A$
                     ; to 0
   ADDQ.L   #4,A0    ;Advance the dest buffer
                     ; pointer (A0) to the first
                     ; data byte of A$
   MOVE.L   ScanPointer,A2;(Set A1 to the scan
                     ; pointer)
   MOVE.L   ScanCounter,D2;If the scan counter (D2)
   BEQ      Scanw_9 ; is zero, then jump to Scanw_9
Scanw_1
   MOVE.B   (A2)+,D0;Fetch the character at which
                     ; the scan pointer points,
                     ; and advance the scan pointer
   SUBQ.L   #1,D2    ;Decrement the scan counter
   CMPI.B   #'!',D0 ;If the character is visible,
                     ; then jump to Scanw_2
   BLT      Scanw_3
   CMPI.B   #$7F,D0
   BLT      Scanw_2
Scanw_3
   TST.L    D2       ;If the scan counter is zero,
   BEQ      Scanw_9 ; then jump to Scanw_9
   BRA      Scanw_1 ;Jump to Scanw_1
Scanw_2
   MOVE.B   D0,(A0)+;Place the character into A$,
                     ; and advance the dest buffer
                     ; pointer
   ADD.L    #1,(A1) ;Increment the current usage
                     ; of A$
   CMP.L    (A1),D1 ;If the current usage of A$ =
                     ; the max length of A$,
   BEQ      Scanw_9 ; then jump to Scanw_9
   TST.L    D2       ;If the scan counter is zero,
   BEQ      Scanw_9 ; then jump to Scanw_9
   MOVE.B   (A2)+,D0;Fetch the character at which
                     ; the scan pointer points,
                     ; and advance the scan pointer

   SUBQ.L   #1,D2    ;Decrement the scan counter
   CMPI.B   #'!',D0 ; If the character is visible,
   BCC      Scanw_2 ; then jump to Scanw_2

Scanw_9
   MOVE.L   A2,ScanPointer;(Update the scan
                     ; pointer)
   MOVE.L   D2,ScanCounter;(Update the scan
                     ; counter)

   MOVEM.L (A7)+,D0-D2/A0-A2;(Pop registers)

   RTS               ;Return


ItoA_                ;BASIC Fn: A$ = STR$(I)
                     ;           (A0)      (D0)

;This routine converts a signed integer from
; binary to a decimal ASCII string

   MOVEM.L D0-D4/A0-A4,-(A7);(Push registers)

   MOVE.L   #ItoA_values,A1;Set the table pointer
                     ; (A1) to the beginning of
                     ; the table
   MOVEQ.L #10,D1    ;Set the table counter (D1) to
                     ; the length of the table
   MOVEQ.L #0,D4     ;Set the character counter (D4)
                     ; to 0
   MOVE.L   (A0)+,D2;Set the limit value (D2) to
                     ; the max length of A$
   MOVE.L   A0,A2    ;(Make A2 point to the current
                     ; usage of A$)
   ADDQ.L   #4,A0    ;(Make A0 and A4 point to the
                     ; first data byte in A$)
   MOVE.L   A0,A4
   TST.L    D0       ;If I >= 0, then jump to ItoA_1
   BPL      ItoA_1
   MOVE.B   #'-',(A0)+;Place a '-' in A$, and
                     ; advance the A$ pointer
   ADDQ.L   #1,D4    ;Increment the character
                     ; counter
   CMP.L    D2,D4    ;If the character counter
                     ; equals the limit value,
   BEQ      ItoA_5   ; then jump to ItoA_5
   NEG.L    D0       ;Complement I
ItoA_1
   CMP.L    (A1),D0 ;If I >= the table value,
   BGE      ItoA_2   ; then jump to ItoA_2
   ADDQ.L   #4,A1    ;Advance the table pointer
   SUBQ.L   #1,D1    ;Decrement the table counter
   CMPI.L   #1,D1    ;If the table counter is not
                     ; yet 1,
   BNE      ItoA_1   ; then jump to ItoA_1
ItoA_2
   MOVE.B   #'0',(A0);Set the character to '0'
   ADDQ.L   #1,D4    ;Increment the character
                     ; counter
ItoA_3
   SUB.L    (A1),D0 ;Subtract the table value
                     ; from I
   BMI      ItoA_4   ;If I is negative, then jump to
                     ; ItoA_4
   ADDQ.B   #1,(A0) ;Increment the data byte in A$
   BRA      ItoA_3   ;Jump to ItoA_3
ItoA_4
   ADD.L    (A1),D0 ;Add the table value to I,
                     ; making I positive again
   ADDQ.L   #4,A1    ;Advance the table pointer to
                     ; the next value
   SUBQ.L   #1,D1    ;Decrement the table counter
   BEQ      ItoA_7   ;If the table counter is zero,
                     ; then jump to ItoA_7
   ADDQ.L   #1,A0    ;Advance the A$ pointer
   CMP.L    D4,D2    ;If the character counter <=
                     ; the limit value,
   BGT      ItoA_2   ; then jump to ItoA_2
ItoA_5
   MOVE.L   D2,(A2) ;Set the current usage of A$
                     ; to the max length of A$
   SUBQ.L   #1,D4    ;Fill the buffer with asterisks
ItoA_6
   MOVE.B   #'*',(A4)+
   DBRA     D4,ItoA_6
   BRA      ItoA_8   ;Jump to Itoa_8
ItoA_7
   MOVE.L   D4,(A2) ;Set the current usage of A$
                     ; to the character count
ItoA_8
   MOVEM.L (A7)+,D0-D4/A0-A4 ;(Pop registers)
   RTS               ;Return

ItoA_values
   DC.L      1000000000
   DC.L       100000000
   DC.L        10000000
   DC.L         1000000
   DC.L          100000
   DC.L           10000
   DC.L            1000
   DC.L             100
   DC.L              10
   DC.L               1


AtoI_                ;BASIC Fn: I = VAL(A$)
                     ;           (D0)   (A0)

;This routine converts a decimal ASCII string to
; a signed binary integer

   MOVEM.L D1-D4/A0-A1,-(A7) ;(Push registers)

   MOVEQ.L #0,D0     ;Set the character field to
                     ; zero
   MOVEQ.L #0,D2     ;Set the result (D2) to 0
   MOVEQ.L #0,D3     ;Clear the negative flag (D3)
   ADDQ.L   #4,A0    ;(Make A1 point to the current
                     ; usage of A$)
   MOVE.L   A0,A1
   MOVE.L   (A0)+,D1;Set the byte counter (D1) to
                     ; the current usage of A$
                     ;(Make A0 point to the first
                     ; data byte in A$)
   BEQ      AtoI_9   ;If the byte counter is zero,
                     ; then jump to AtoI_9
   MOVE.B   (A0)+,D0;Fetch the first data byte in
                     ; A$ (Make A0 point to the
                     ; second data byte in A$)
   CMPI.B   #'-',D0 ;If the byte is not a '-', then
   BNE      AtoI_1   ; jump to AtoI_1
   MOVEQ.L #1,D3     ;Set the negative flag
   BRA      AtoI_4   ;Jump to AtoI_4
```

(continued)

```
AtoI_1
   CMPI.B  #'0',D0 ;If the byte is not numeric,
   BLT     AtoI_9  ; then jump to AtoI_9
   CMPI.B  #'9',D0
   BGT     AtoI_9
   LSL.L   #1,D2   ;Multiply the result by 10
   MOVE.L  D2,D4
   LSL.L   #2,D2
   ADD.L   D4,D2
   SUBI.B  #'0',D0 ;Convert the byte to binary
   ADD.L   D0,D2   ;Add the byte to the result
AtoI_4
   SUBQ.L  #1,D1   ;Decrement the byte counter
   BEQ     AtoI_9  ;If the byte counter is zero,
                   ; then jump to AtoI_9
   MOVE.B  (A0)+,D0;Fetch the next data byte in A$
                   ;(Make A0 point to the next
                   ; data byte in A$)
   BRA     AtoI_1  ;Jump to AtoI_1
AtoI_9
   TST.L   D3      ;If the negative flag is clear,
   BEQ     AtoI_A  ; then jump to AtoI_A
   NEG.L   D2      ;Negate the result
AtoI_A
   MOVE.L  D2,D0   ;Set D0 to the result

   MOVEM.L (A7)+,D1-D4/A0-A1 ;(Pop registers)

   RTS             ;Return

   IFND    ScanPointer
ScanPointer
   DS.L    1
   ENDC
   IFND    ScanCounter
ScanCounter
   DS.L    1
   ENDC

   IFND    ConIn
ConIn
   DS.L    1
   ENDC

   IFND    ConOut
ConOut
   DS.L    1
   ENDC

   IFND    DosLibraryHandle
DosLibraryHandle
   DS.L    1
   ENDC

   IFND    SystemSP
SystemSP
   DS.L    1
   ENDC
```

## *Listing 4*
## *Sample Runs*

```
1>test 12 23 34 45 56 67 78 89 90
-12- 12
-23- 23
-34- 34
-45- 45
-56- 56
-67- 67
122334455667
The total is: 237

1>test   12  -3  4r5  -33xx hello,world 55
-12- 12
--3- -3
-4r5- 4
--33xx- -33
-hello,world- 0
-55- 55
12-34-33055
The total is: 35

1>test 1234567890 0987654321 00324 0345tt
-1234567890- 1234567890
-0987654321- 987654321
-00324- 324
-0345tt- 345
-- 0
-- 0
123456789098765432132434500
The total is: -2072744416
1>
```

•AC•

**MOVING?**

**SUBSCRIPTION PROBLEMS?**

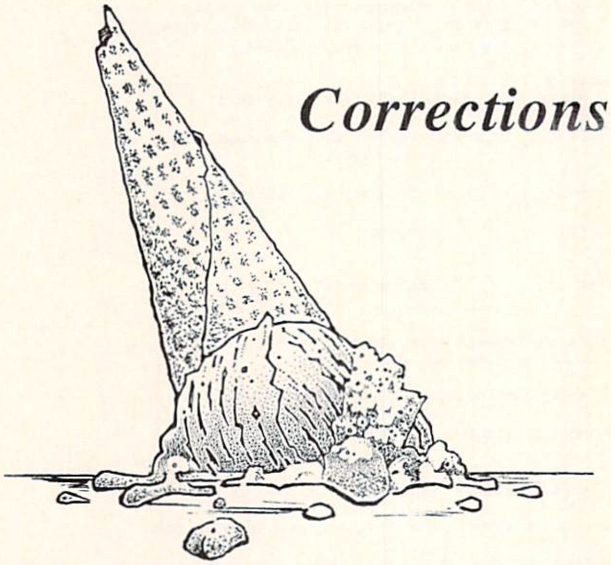
Please don't forget to let us know.
If you are having a problem with your subscription or if you are planning to move, please write to:

**Amazing Computing Subscription Questions**
**PiM Publications, Inc.**
**P.O. Box 869**
**Fall River, MA 02722**

Please remember, we cannot mail your magazine to you if we do not know where you are.

*Please allow four to six weeks for processing.*

## *Oops ...*

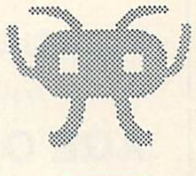


## *Corrections*

Our *AiRT Review* , published in AC V3.1, referred to three "Figures." We planned to show some original features of PDJ Software's innovative programming package. Regrettably, the figures were inadvertently omitted from the final copy. We apologize for any inconvenience caused by our error.—Ed

# In the Public Domain

*by C.W. Flatte*



The Fred Fish Public Domain Library continues to grow by bytes and bounds. Ten disks have been added to the collection, topping it out at 128 disks! There are many highlights (don't miss Amoeba Invaders, the Space Invaders clone on FF 120), hot-spots, tid-bits, and delights on the new entries, so let's go.

### Fred Fish #119

This disk contains version 3.9e of Daniel Lawrence's variant of Dave Conroy's MicroEMACS. This is an update of the version released on Fred Fish #93. On this disk only: the first appearance of extensive documentation on disk.

New features since version 3.8i (on Fred Fish #93) include a new standard startup file, twenty new variables, three new directives, and sixteen new functions. Also provided: an on-line "learn by doing" tutorial.

Both Lattice C and Manx C compiled versions are provided as well as the source and documentation for both versions. NOTE: In order to get all the source and documentation files to fit on a single disk, Fred had to archive their contents with Zoo, which compresses the files about 30%-50%. (Zoo can be found on Fred Fish #108.) The executables and on-line tutorial are not archived, and can be run from the Workbench.

### Fred Fish #120

**Amoeba Invaders**

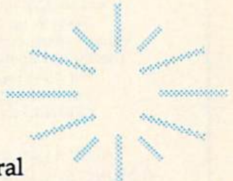
This clone of the original Space Invaders is one of the best freeware games for the Amiga to date. Unlike many of the commercial games, it even works correctly in a multitasking environment. THIS IS A GREAT GAME! It definitely qualifies as a don't-miss!!! It's not even shareware (you'll probably feel guilty, not sending the authors a donation... so send one anyway).

**BackGammon V1.0**

This graphical backgammon game was written by author Robert Pfister for an undergraduate Artificial Intelligence course project. Includes the source in Aztec C. Includes a help screen and allows you to back up or redo a move. Only allows play against the computer. Backgammon, anyone?

**Bank'n V1.3**

This is a complete checkbook management system by programmer Hal Carter. Comprised of eight separate programs, the Bank'n system will allow you to keep track of your checking account with ease. The system boasts user-customized expense code tables, as well as several detail and summary reports. Also has the ability to search the checkbook journal for a desired argument. The screens are pleasantly designed and easy to use and understand. Includes a very nice instructional program.

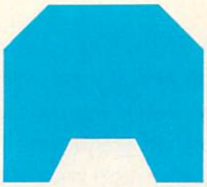


**EgyptianRun V1.1**

The pyramids have come alive and are out to take over the world! You are the world's only hope! You have to maneuver your jeep past the oncoming missiles to get within firing range of the control pyramid. Nice graphics and smooth animation. Also good sound. By Chris Hames of Australia. The source, in C, is available from the author for a $10.00 donation.

**IconImage**

A program to replace an old image with a new image, without affecting icontype, drawer data, etc. A useful tool for manipulating icon images without changing the other parameters in the diskobject structure. By Denis Green. Includes the source in C.



### Fred Fish #121

**BasicStrip**

An AmigaBASIC program by George Trepal that helps convert programs written in other forms of BASIC to AmigaBASIC. It deletes line numbers, puts in its own jump labels, and tries to format the result as a structured program.

*(continued)*

Conflict Recreations, Inc.

*PRESENTS*

**AGE OF SAIL**

**Age of Sail** is the first of its kind in warfare simulation. Centered around 17-19th century sail powered warships, true renditions of classic naval battles will be reenacted.

**Age of Sail** is a multiplayer game that allows up to 40 ship captains to play via electronic bulletin boards (BBS), direct connect modem, or using one computer. Designed for play on differing computers, ASC II files with game data can be sent to anyone, anywhere via modem.

**Age of Sail** faithfully recreates sailing allowing one degree turns and speed changes of one knot. Positions are calculated with 64 bit accuracy to ensure ship movement even when drifting in low velocity winds.

Grapple up to four ships, assign boarding parties, and capture enemy ships, direct your gun captains in loadouts for firing shot. Give them their targets and let the broadside commence.

For 2-40 players.
Requires workbench 1.2, kickstart 1.2, and 512k.
Simulations for serious gamers.

*NOW AVAILABLE FOR THE COMMODORE-AMIGA A-1000.*
*COMING SOON FOR THE APPLE MACINTOSH AND THE ATARI ST.*

**PRICE $39.95 + $3.00 SHIPPING AND HANDLING.**
*CONNECTICUT RESIDENTS INCLUDE SALES TAX.*
*ALLOW 4 TO 6 WEEKS FOR DELIVERY.*

*SEND CHECK OR MONEY ORDER TO:*

Conflict Recreations, Inc.
P.O. Box 272
Oakdale, CT 06370

*WE SUPPORT ON-LINE GAMING VIA:*
*COMPUSERVE: 72375, 225 PLINK: SILVER MAC BIX: GMcLEAN*
DEALER INQUIRIES ACCEPTED.

**DataPlot**

DataPlot is a shareware plotting system, written in AmigaBASIC, by Dale Holt. It includes a least squares curve fitting program and many sample plot files. Printer output is supported.

**Plot**

A shareware 3D graphing program, written in AmigaBASIC, by George Trepal. This program draws 3-D graphs as either wire frames or colored solids, with hidden lines eliminated. Includes three sample plot files. The source is available from the author.

**Stairs**

This AmigaBASIC program by Gary Cuba demonstrates a musical illusion based upon perceptual circularity of widely spaced tones whose volumes are defined as a sinusoidial relationship to their frequency. (Say that three times fast.) Will the tones ever stop ascending?

**Uedit V2.3**

Uedit is a very nice editor by Rick Stiles. Uedit's features include a learn mode, a command language, menu customization, and other user configurable features. An update to Fred Fish #60.

**WBColors**

A little program by Stefan Lindahl of Sweden to change the Workbench colors to a predetermined color set. Can be used for programs that expect to be booted off their distribution disk, but instead are run from a hard disk. Includes the source in C.

## Fred Fish #122

**Asteroids**

Asteriods is a nicely done "Asteriods-clone" game by Rico Mariani. A unique feature of this version is the end user's ability to replace the images (a single ILBM file) and sounds (8SVX sound files). Nice job, Rico!

**Iff2Pcs V1.0**

An interactive puzzle program, by Ali Ozer, which will take any IFF file (containing up to 16 colors), and break it up into squares to create a puzzle which the user can then piece back together. This is an outstanding program. Kudos to Ali Ozer. Includes the source in Manx C.

**Names**

Names is a generic shareware mailing list manager created by Ernie Nelson. It's a little bit rough right now, but it'll get the job done.

**Pr**

Similar to the Unix "pr" program, this little utility by Samuel Paolucci prints listings in different formats. Includes the source in C.

**Pushover**

Here is an original board strategy game, written in AmigaBASIC by Russ Yost. The object of the game is to push your pieces on the board until you get five in a row in any direction. Includes the source in AmigaBASIC.

**PuzzlePro v1.0**

Written in AmigaBASIC, this program by Syd Bolton creates a puzzle from an IFF picture, which the user can then piece back together. Please note, only the binary is included. The source is available from the author.

## Fred Fish #123

**Arp**

ARP is an acronym for "AmigaDOS Replacement Project." Arp is an effort led by Charlie Heath of Microsmiths, Inc., to replace the current DOS in a compatible fashion, so current programs will continue to work. Arp also improves DOS wherever possible in an effort to alleviate some of its downfalls. Check it out!

**Car**

Allen Hasting does it again with this entry to the Badge Killer Demo Contest. It is apparently an inside joke relating to a well-known Amigan's experience with a certain high-end graphics hardware manufacturer. Please note, you must have at least 1Mb RAM to view the entire loop. Good stuff...

### Fred Fish #124

**Icons**

On this disk are two drawers full of animated icons by L. Pfost. Some are really good! (Some are really bad.) You may just find an icon for that little program you have been meaning to "Workbenchize."

**Tarot**

Tarot is an AmigaBASIC program written by L. Pfost as an exercise for learning AmigaBASIC. It contains some very well done renditions of tarot cards. A good learning tool.

### Fred Fish #125

**ElGato**

OH NO! More entries to the Badge Killer Demo Contest! This time Kevin Sullivan flaunts his talent. Some of the most incredible animation ever! This demo even includes a background music arrangement, which requires Sonix to use. The workbench icons are awesome! Please note, this demo requires an extra 256K of FAST RAM to run in its entirety.

### Fred Fish #126

**Colour**

This IFF file color manipulation program by John Russell allows a user to save color sets to data files, load color sets from data files, or interactively change the colors. Includes the source in C.

**Dance**

More entries to the Badge Kill Demo Contest. These two programs, "dancing polygons," are John Olsen's entries. They are variations of one another; however, they do demonstrate the range of colors available on the Amiga. Includes the source.

**HBHill**

This animation is Kevin Sullivan's entry to the Badge Killer Demo Contest. It is the first known animation using the Amiga's "Extra Half Brite" mode.

**Iconify**

Iconify is a subroutine by the infamous Leo Schwab that creates an icon on the Amiga screen that subsequently can be dragged around, and double-clicked on. You can use this to have your programs "iconify" themselves to temporarily get out of the user's way. Includes source and demo program in C.

**OnlyAmiga**

This animation is Iqbal Singh Hans' entry to the Badge Killer Demo Contest. It consists of three balls being juggled by pyramids rotating on their tops. Freaky!

**Suplib**

IT'S HERE! Finally the Matt Dillion support library needed to rebuild various programs of Matt's from the source, including DME, DTERM, etc. It even includes the source! Great!

**VCheck V1.2**

This virus detection program is a very important piece of software. Viruses DO CAUSE DAMAGE! This virus detector was developed by Bill Koester of Commodore Amiga Technical Support. This version tests for a virus in memory, or on specific disks.

Oh no! Gotta go! See you next time for disks 127 and 128. Until then ... tell 'em CW sent you!

•AC•

DYNAMIC DRUMS V 1.1

The program that transforms your Amiga™
into a professional drum machine.

- Incredibly realistic sound
- Create your own studio-quality drum tracks
- Real or step-time programming
- Graphic Editing
- Over 100 percussion samples included or use your own IFF samples
- Fully adjustable volume and tuning levels
- Randomizing options for a dynamic, human feel
- Improved MIDI Implementation
- Velocity sensitive using external MIDI device

Requires 512K Amiga™ DEALER INQUIRIES INVITED
MI, FL & CA add sales tax
See your Local Dealer or Send Check or Money Order for $79.95 to:

New Wave
SOFTWARE

P.O. Box 438, St. Clair Shores, MI 48080, (313) 771-4465
Amiga is a trademark of Commodore-Amiga Inc.

# The A.M.U.G. BBS List

## **514** BBS Phone Numbers in U.S. and Canada

***compiled by:*** Joe Rothman, Chet Solace, and Dororthy Dean

This is the A.M.U.G. list of all known Amiga-supporting BBSs throughout the U.S. and Canada. It was compiled as a cooperative effort by the BBS Sysops listed below. All listings with a "C" next to the phone number have been confirmed by a Sysop or another reliable source. If a "B" appears in the Status column, we got a busy signal every time we called the number.

Each Sysop has his own way of formatting. Chet's list is formatted as the phone directory of AMIC PD Term by David Salas and the new 16 Color Access! by Keith Young. Dorothy and Chet both list Foreign Amiga BBSs and track PC Pursuit compatibility. But, we all have one thing in common: We like to spread the word about Amiga telecommunications and PD. The latest versions of our respective lists are available for Download or Capture at the following BBSs:

| | | |
|---|---|---|
| *Joe Rothman* | | |
| **A.M.U.G.** | 516-234-6046 | BBSlist |
| *Chet Solace* | | |
| **Amiga America** | 619-364-3816 | The Final List |
| *Dorothy Dean* | | |
| **PSA-BBS** | 414-278-5390 | PSA-BBSs |

A.M.U.G.'s BBSlist is now thoroughly confirmed. A modem connection was made at least once on every number and someone was able to complete a logon. All previously unconfirmed numbers have been either confirmed or removed from the list.

Remember, any number can become inactive. Use the Status line to track any changes in operational status. If you find any changes in the name, phone number, operational status, hours, baud rate, or features of these BBSs, or if you are able to provide additional numbers or information, please contact us and leave a message with the information. Your cooperation is appreciated. We know more Amiga boards are out there; help us find them.

All added or changed numbers are marked with a * beside the phone number.

Delete or discard old copies of these lists; some numbers have been changed to voice lines by the phone company.

[NOTE: The format of the list is as follows: State, BBS Name, Phone Number, Status, Hours, Maximum Baud Rate, and Special Notes.]

## USA

| | | | | | | |
|---|---|---|---|---|---|---|
| AK | Northern Lights | 907-337-4136 | C | 24 HOUR | 24 | 1st Alaska Color |
| AK | Northern Node | 907-452-1460 | C | 24 HOUR | 12 | Amiga Echomail |
| AK | Millenium Falcon | 907-372-5345 | C | 24 HOUR | 12 | Amiga/Color |
| AL | The System | 205-349-2951 | C | 24 HOUR | 3 | Messages only |
| AL | Lighthouse | 205-553-5322 | C | 24 HOUR | 12 | 20 Mb, Amiga Section |
| AL | Smorgasboard | *205-745-3989 | C | 9PM-6PM | 12 | Amiga Section |
| AL | Bloom County | *205-772-8526 | C | 24 HOUR | 24 | 30 Mb, Amiga Section |
| AL | Magnolia | 205-854-6407 | C | 24 HOUR | 24 | 60 Mb, Amiga Section |
| AL | Connection | 205-854-9074 | C | 24 HOUR | 24 | Amiga Section |
| AR | Thunder BBS | 501-329-8370 | C | 24 HOUR | 12 | Amiga Section |
| AR | C.C.X. BBS | 501-450-1400 | C | 24 HOUR | 12 | Collie, Amiga Section |
| AR | C.I.A. BBS | 501-470-0624 | C | 24 HOUR | 12 | C128, Amiga Section |
| AR | UBBS (U of AR) | 501-568-9464 | C | 24 HOUR | 12 | Vax, Huge Amiga section |
| AR | Next Generation | 501-666-0004 | C | | 24 | Amiga |
| AZ | Ragarm 128 | 602-230-1463 | C | | 12 | C128, Amiga Section |
| AZ | Bob's Ans. Mach. | 602-242-3158 | C | | 12 | Amiga Section |
| AZ | MasterCom | 602-244-8096 | C | 24 HOUR | 24 | BBS-PCI 30 Mb |
| AZ | G & G BBS | *602-258-9377 | C | 24 HOUR | 12 | Amiga Section |
| AZ | Golden Citadel | 602-486-8750 | C | | 12 | C64, Amiga Section |
| AZ | Cafe Americana | *602-493-1280 | C | | 12 | |
| AZ | AmyHolics | *602-582-5174 | C | 24 HOUR | 24 | |
| AZ | Daemon's Den | *602-841-0509 | C | 24 HOUR | 12 | 20 Mb Amiga Section |
| AZ | Lost Horizon | 602-849-0110 | C | | 12 | C128, Amiga Section |
| AZ | Cornerstone Manor | *602-935-6519 | C | | 12 | C64, Amiga Section |
| AZ | Space Station | 602-978-4430 | C | | 12 | |
| AZ | Guru's Hut | *602-992-0169 | C | | 12 | |
| CA | Amiga Techniques | *209-436-1643 | C | | 24 | RBBS/BIG Amiga Section |
| CA | Amiga Group Enthu | 209-438-0510 | C | 24 HOUR | 12 | Commlink |
| CA | KPS Micro BBS | 209-781-1344 | C | Mid-10AM | 24 | 60 Mb, Amiga Section 8N1 |
| CA | Tele-Web | 213-372-4050 | C | | 12 | Amiga Section |
| CA | Aegis Software | *213-399-7316 | C | 24 HOUR | 12 | Aegis Developement BBS |
| CA | Compu-Center | 213-421-1041 | C | | 12 | Amiga Messages Only |

| | | | | | | |
|---|---|---|---|---|---|---|
| CA | Access-1 | 213-537-3378 | C | | 24 | 40 Mb, PW: Genius |
| CA | Machine Do | 213-548-3546 | C | | 12 | Fido, Amiga section |
| CA | L.A.A.U.G. | 213-559-7367 | C | | 12 | Los Angeles Amiga User Gr |
| CA | New Age BBS | 213-559-4948 | C | 9PM-7AM | 12 | Tag BBS, Amiga |
| CA | Bilbo's Hideaway | 213-640-6104 | C | | 12 | Tag BBS |
| CA | ADS-BBS | *213-823-1622 | C | 24 HOUR | 12 | Store TAG BBS |
| CA | Custom BBS | *213-839-6867 | C | | 12 | Amy/Devel. Was Flying Bug |
| CA | Cretans Corner | 213-861-3830 | C | 24 HOUR | 12 | 20 Mb, 8N1, No time limit |
| CA | Amiga Temple | *408-241-7802 | C | | 24 | Amiga, BBS-PC |
| CA | Shark's Head | 408-247-4810 | C | | 12 | Unix with Amiga Sig |
| CA | Shrink Tank | 408-257-0323 | C | | 12 | C= Lib, Some Amiga |
| CA | Refuge | 408-262-3963 | C | | 12 | Amiga Section |
| CA | Developers Xchg. | 408-372-1722 | C | 6PM-8AM | 12 | MaxiCorp, 24 Hrs Wkends |
| CA | The Exchange | *408-659-3078 | C | | 12 | Fido, Amiga |
| CA | PortalSys.1 | *408-725-0561 | C | 24 HOUR | 12 | Usenet access |
| CA | PortalSys.2 | *408-725-0569 | C | 24 HOUR | 24 | Usenet access |
| CA | HT-BBS | 408-737-0900 | C | 7PM-8AM | 12 | Run by store |
| CA | Digtal Mail box | *408-946-6154 | C | 24 HOUR | 24 | Amiga Section |
| CA | Ami Addicts | *415-222-9416 | C | | 12 | User Group BBS |
| CA | THE WELL | 415-332-6106 | C | | 12 | Rates $2/H + $8/Mo |
| CA | Amiga West | 415-355-7162 | C | 24 HOUR | 12 | 20 Mb |
| CA | Records Dept | 415-426-0470 | C | | 12 | Fido, Echomail |
| CA | B.A.C.C. | *415-455-0410 | C | 24 HOUR | 24 | Amiga section |
| CA | Amiga Connection | 415-562-7097 | C | 24 HOUR | 24 | |
| CA | Dream Machine | 415-581-3019 | C | 24 HOUR | 24 | 80+Mb, Amiga Section |
| CA | Missing Link | *415-581-9452 | C | | 12 | RBBS Big Amiga Section |
| CA | F.A.U.G. | 415-595-2479 | C | 6PM-7AM | 12 | 24 Hrs. on weekends |
| CA | RSVP-BBS | 415-659-9169 | C | | 12 | Fido, Amiga Section |
| CA | Nerd's Nook | 415-672-2504 | C | | 24 | Opus, Fido, Amiga Section |
| CA | Enterprise | *415-895-1788 | C | | 12 | Amiga Section |
| CA | BBS-JC | 415-961-7250 | C | 24 HOUR | 24 | Amiga section |
| CA | Computer Outlet | *619-282-6815 | C | | 24 | Eves/Weekends Amiga Sec. |
| CA | El Amigo | *619-299-9804 | C | 24 HOUR | 24 | BBS-PCI |
| CA | Desert Amiga Bd | 619-323-1522 | C | 24 HOUR | 12 | BBS-PCI |
| CA | Amiga America | 619-364-3816 | C | 24 HOUR | 24 | 25 Mb, Color use 8N1 |
| CA | SD Midi Net | 619-452-0097 | C | | 12 | Amiga Echomail |
| CA | Prides XXing | *619-464-6271 | C | 24 HOUR | 12 | Amiga Section |
| CA | Far Star 1 | *619-480-9686 | C | | 12 | Amiga = Section 7 |
| CA | Mouse Trap | *619-582-7447 | C | | 24 | BBS-PC |
| CA | Computer Blvd | *619-589-0565 | C | 24 HOUR | 24 | Amiga Section |
| CA | Computers Plus | *619-691-7862 | C | 24 HOUR | 24 | Amiga Section |
| CA | Far Star 2 | *619-726-4419 | C | | 24 | Amiga = Section 7 |
| CA | Antarctica | 619-758-3827 | C | | 12 | Tag-BBS, Color, 2 Drives |
| CA | AMIC 2000 B | *707-579-0522 | C | 24 HOUR | 24 | Huge Library, Color |
| CA | AMIC 2000 A | 707-579-0523 | C | 24 HOUR | 24 | Huge Library, Color |
| CA | The Tarot | *707-833-1805 | C | | 12 | |
| CA | Coast BBS | 707-964-7114 | C | 24 HOUR | 12 | Erase 1st O at Prompt |
| CA | C.U.T.E. | 714-532-5698 | C | 24 HOUR | 12 | Text files only |
| CA | King James | 714-537-7355 | C | | 24 | Fido, New Name |
| CA | The Black Hole | 714-543-6337 | C | 24 HOUR | 24 | 40 Mb, 8N1 |
| CA | Mike's Rainbow | *714-544-3369 | C | | 24 | Fido, Amiga Section |
| CA | Black Chamber | 714-646-2723 | C | 24 HOUR | 24 | 20 Mb, |
| CA | Dragon Soft | 714-671-7831 | C | 24 HOUR | 24 | BBS-PCI |
| CA | Misty Mountain | *714-684-6553 | C | | 12 | Mainly Amiga |
| CA | Amiga Line | 714-772-4097 | C | 24 HOUR | 12 | Color Many files |
| CA | 77-Amiga | 714-772-6442 | C | 24 HOUR | 24 | 30 Mb |
| CA | Soft Celler | 714-772-9671 | C | | 12 | Amiga Section |
| CA | New Age BBS | *714-822-2983 | C | | 12 | Amiga's 1st Metaphys.bbs |
| CA | Fontana | 714-823-3673 | C | 24 HOUR | 24 | Amiga Section |
| CA | Point After | *714-826-0125 | C | | 96 | C128, Amiga Section |
| CA | Kingdom 1 | 714-829-8908 | C | 24 HOUR | 12 | BBS-PCI |
| CA | Myco-D | 714-832-8016 | C | | 12 | Amiga Section |
| CA | Learning Place | 714-849-8332 | C | | 12 | Amiga Section |
| CA | Amiga Kingdom 3 | 714-874-3177 | C | 24 HOUR | 12 | |
| CA | Amiga Kingdom 2 | 714-886-0324 | C | 24 HOUR | 12 | |
| CA | Rancho Amiga BBS | 714-945-1020 | C | 24 HOUR | 24 | |
| CA | Smorgasbord | 714-957-0142 | C | | 12 | Amiga Section |
| CA | Light House | 805-272-1812 | C | | 12 | Amiga Section |
| CA | ATB-BBS | 805-499-7151 | C | 24 HOUR | 24 | Name change |
| CA | Ventura TBBS | 805-656-3746 | C | 24 HOUR | 24 | Amiga section |
| CA | Abacus | 805-832-7186 | C | | 12 | $ Membership |
| CA | Amiga Project | 805-834-9383 | C | 24 HOUR | 12 | BBS-PCI on Amiga |
| CA | L.O.I.S. | *805-928-4225 | C | 24 HOUR | 12 | 6line, 24hr Amiga Section |
| CA | Compucation | 805-967-0895 | C | 24 HOUR | 24 | Amiga section |
| CA | 1939 BBS | 818-368-4248 | C | 24 HOUR | 12 | SFVAUG |
| CA | Message Center | 818-353-8437 | C | 24 HOUR | 12 | Amiga Messages Only |
| CA | L D Voyager | 818-354-5065 | C | | 12 | Seadog, Color |
| CA | C.A.B.B.S. | *818-355-1336 | C | | 12 | Wildcat BBS, Amiga Section |
| CA | Computer Lodge | 818-444-9850 | C | 4PM-7AM | 12 | 40 Mb, 24 Hours weekends |
| CA | Infinity | *818-760-0943 | C | | 12 | Amiga section, New Name |
| CA | House Atreides | 818-965-7220 | C | 24 HOUR | 12 | Fido, Amiga Section |
| CA | United Computers | 818-982-8495 | C | 3PM-11P | 24 | BBS-PCI |
| CA | 68000 BBS | *916-351-1890 | C | 24 HOUR | 24 | ST, Amiga Section |
| CA | Hot Tub | 916-689-4670 | C | 24 HOUR | 12 | 'tween 9PM Sun & 8AM Sat |
| CO | Network Unlimited | 303-296-3210 | C | | 12 | Nights only |
| CO | The Buck Board | *303-425-9283 | C | 24 HOUR | 24 | "TelegraphHill" |
| CO | Buck Board | 303-427-9539 | C | 24 HOUR | 24 | TBBS, Amiga Section |
| CO | Green Line | *303-469-2053 | C | | 12 | Amiga Section |
| CO | Boulder Fido | 303-497-6968 | C | | 12 | Amiga Section |
| CO | Opus BBS | *303-544-3704 | C | 24 HOUR | 12 | Amiga Section |
| CO | M.A.E. BBS | 303-564-0618 | C | 24 HOUR | 12 | BBS-PCI on Amiga |
| CO | The Grotto | 303-694-9050 | C | 24 HOUR | 12 | TBBS, Amiga Section |
| CO | Mile High BBS | 303-730-2250 | C | | 12 | |
| CO | Stampede | 303-799-9733 | C | | 12 | ST, Amiga Section |
| CO | Computer Discount | *303-825-0419 | C | | 12 | Store BBS, Amiga Section |
| CO | DBSystem | 303-893-5717 | C | 24 HOUR | 12 | Amiga Section |
| CO | Enterprise | 303-980-1304 | C | | 12 | |
| CO | Unknown TBBS | 303-988-8155 | C | 24 HOUR | 12 | Amiga Section |
| CT | A.I. Expert | *203-263-5783 | C | 24 HOUR | 24 | FIDO, Some Amiga |
| CT | Shoreline | 203-468-2853 | C | | 12 | Fido, Amiga Section |
| CT | Wilton Woods | *203-762-8481 | C | | 24 | Amiga Echomail |
| CT | NORAD | *203-846-3522 | C | | 24 | Amiga Section + EchoMail |
| CT | PCC BBS | 203-848-7373 | C | 24 HOUR | 12 | Amiga with 2 Drives |
| CT | FlagShip | *203-871-1092 | C | 24 HOUR | 24 | Amiga Section |
| CT | The Archives | 203-938-9163 | C | 24 HOUR | 24 | Amiga Section |
| DC | Concourse | 202-328-9737 | C | 24 HOUR | 12 | Amiga Section |
| DC | Amiga Exchange | 202-439-6980 | C | 24 HOUR | 12 | MD Amiga Magazine DC # |
| DC | Computer Ease | *202-856-3372 | C | | 12 | |
| DE | PC-NUG Brd | 302-737-2294 | C | 24 HOUR | 12 | Fido, Amiga Section |
| FL | ORAS | *305-235-6752 | C | 5PM-7AM | 24 | Monday - Friday |
| FL | Vikenholm | *305-258-0862 | C | 5PM-7AM | 12 | Amiga Section |
| FL | Micros Etc. | *305-425-7054 | C | 7PM-10AM | 12 | |
| FL | ARIA | 305-435-9837 | C | 24 HOUR | 24 | |
| FL | Fortune Cookie | 305-439-2136 | C | 12P-8AM | 24 | Amiga supported |
| FL | Commodore Culb | *305-478-3008 | C | | 12 | LOCAL user group on C64 |
| FL | Nervous System | *305-554-9332 | C | | 12 | Amiga Section |
| FL | NiteOwl | 305-588-0487 | C | 24 HOUR | 24 | RBBS, Amiga Section, Color |
| FL | AmigaLink | *305-622-7049 | C | 24 HOUR | 12 | Video, Graphics, Animation |
| FL | Dark Castle | *305-624-6123 | C | | 12 | Amiga Section |
| FL | Cocoa Beach Onlin | 305-636-2664 | C | | 24 | Spacecoast Amiga Users |
| FL | The Bug Shop | 305-641-1162 | C | 24 HOUR | 24 | Computer Flea Market |
| FL | DOS BBS | 305-737-1644 | C | | 12 | |
| FL | Twilight Board | 305-738-7539 | C | 24 HOUR | 12 | 2 Drives |
| FL | Doctor Fido | 305-744-7862 | C | | 24 | Amiga Section |
| FL | MSS Nightline | 305-790-0773 | C | 5PM-9AM | 24 | New # 24 Hrs on Weekends |
| FL | MSS-HQ | 305-790-0774 | C | 24 HOUR | 96 | # Micro Systems Software |
| FL | Amiga Cave | 305-798-3576 | C | 8PM-7AM | 12 | |
| FL | Amiga Cave 2 | 305-798-8959 | C | 24 HOUR | 12 | |
| FL | Micros Etc. | 305-831-3241 | C | 7PM-10AM | 12 | |
| FL | Commodore Forum | 305-832-7369 | C | | 12 | |
| FL | Tampa Ami Xchg | *305-869-0226 | C | 24 HOUR | 12 | BBS-PCI, New Name |
| FL | IntuiLand | *813-293-3926 | C | | 24 | BBS-PCI, Amiga |
| FL | Sun Coast | *813-351-8050 | C | | 24 | Amiga echomail |
| FL | Iron Age BBS | 813-527-2616 | C | | 12 | BBS-PCI, Amiga Section |
| FL | The Wave | *813-579-8037 | C | 24 HOUR | 12 | Amiga |
| FL | Pleasuredome 1 | 813-799-3191 | C | 24 HOUR | 24 | Multi line 72 Mb |
| FL | Amiga-BBS | 813-924-2626 | C | 24 HOUR | 12 | Run by Store |
| FL | Pleasuredome 2 | 813-942-1975 | C | 24 HOUR | 24 | Multi line 72 Mb |
| FL | T. A. X. | *813-932-2149 | C | 7PM-7AM | 12 | 24 Hours Weekends |
| FL | "Mon Ami" | 813-985-7624 | C | 9PM-7AM | 12 | |
| FL | The Hobbit Hole | 904-243-6219 | C | 24 HOUR | 24 | BBS-PCI, 50 Mb |
| FL | DataWest | 904-262-9629 | C | | 12 | C64, Amiga Section |
| FL | Short Circuit | 904-581-1510 | C | 24 HOUR | 12 | BBS-PCI |
| FL | Commodore Xch | 904-651-8908 | C | | 12 | C64, Amiga Section |
| FL | Puppy Power | 904-729-1767 | C | 24 HOUR | 12 | BBS-PCI on Amiga |
| FL | Casa Mi Amiga | 904-733-4515 | C | 24 HOUR | 12 | Fido, All Amiga |
| FL | Socrates | 904-771-7140 | C | | 12 | Amiga Section |
| GA | MetalLand | *404-327-2327 | C | 24 HOUR | 24 | Amiga Messages |
| GA | Comm.Corner | *404-363-0105 | C | | 12 | Amiga Section |
| GA | Sanctuary | *404-682-7033 | C | 24 HOUR | 12 | Amiga Messages |
| GA | Index II | *404-991-3569 | C | 24 HOUR | 12 | Amiga Section |
| GA | Index I | *404-961-7386 | C | 24 HOUR | 12 | Amiga Section |
| GA | Tardis | 912-247-2194 | C | 24 HOUR | 12 | |
| GA | White Tower | 912-283-9148 | C | 24 HOUR | 12 | |
| GA | VSC-RBBS | 912-333-5975 | C | 24 HOUR | 24 | Color, Amiga Section |
| HI | Cyber Systems | 808-521-3306 | C | 24 HOUR | 24 | Amiga, BBS-PCI |
| IA | The City | 515-283-1902 | C | 24 HOUR | 24 | BBS-PCI on Amiga |
| IA | Micro-Mack | 515-294-3977 | C | | 24 | Amiga Section |
| IA | Amiga Zone | 712-366-9747 | C | 24 HOUR | 12 | Tag BBS (PD) |
| IL | Anonymous | *217-333-8301 | C | 24 HOUR | 12 | Fido, Amiga Section |
| IL | Champagne FIDO | *217-359-3431 | C | 24 HOUR | 24 | Amiga Section |
| IL | Tranquility II | *217-384-8173 | C | 24 HOUR | 24 | Messages only |
| IL | Gallifrey Alt. | *309-797-8160 | C | 24 HOUR | 12 | Amiga Section |
| IL | Aliens BBS | 312-257-7747 | C | 24 HOUR | 24 | Ram:,Color,Animation |
| IL | Blue Thunder | 312-289-1089 | C | 24 HOUR | 24 | 30 Mb, Amiga Section |
| IL | Amiga Doc | 312-351-8815 | C | 24 HOUR | 24 | |
| IL | Computer Sales 2 | 312-359-3431 | C | 24 HOUR | 24 | Second Line |
| IL | Peeping Tom | *312-543-8112 | C | | 24 | BBS-PCI on Amiga |
| IL | Gallifrey ]I[ | 312-635-8763 | C | | 12 | |
| IL | BBS 1984 | 312-841-2401 | C | 24 HOUR | 24 | Color, Amiga Support |
| IL | BBS Chicago | 312-842-1745 | C | 24 HOUR | 24 | Animated Color |
| IL | A.S.U.G. BBS | 618-265-3113 | C | 9AM-9PM | 12 | Saturday & Sunday Only! |
| IL | Wovsed 200 | 618-378-2133 | C | 5PM-8AM | 24 | Use Username:GUEST |
| IL | Pegasoft | 815-284-3681 | C | 3 Days | 12 | Mon, Wed & Fri 7PM-10PM |
| IL | Hacker's Tele Rag | 815-935-5480 | C | 24 HOUR | 12 | C64, 20 Mb w/Amiga Sec |
| IN | C.P.U. Inc | *317-546-4944 | C | 6PM-10AM | 24 | 33 Mb, 24 Hours Weekends |
| IN | Fort Knox | *317-962-1244 | C | | 24 | BBS-PCI, 2 drives |
| KS | Amig-Oz | 316-283-9210 | C | 24 HOUR | 24 | BBS-PCI Ram: 3 Drive |
| KS | J & L Electonics | 316-624-8068 | C | 6PM-9AM | 24 | Run by store |
| KS | Datalink | 316-651-0365 | C | 24 HOUR | 24 | BBS-PCI On Amiga |
| KS | Scherer's Place | *316-683-3660 | C | 24 HOUR | 24 | Amiga Section |

| | | | | | | |
|---|---|---|---|---|---|---|
| KS | Wichita BBS | *316-943-6030 | C | 24 HOUR | 12 | Amiga Msgs only |
| LA | Delta Designs | 504-456-6334 | C | | 12 | 24 Hours Sunday-Thursday |
| LA | Fire Scene | 504-641-4789 | C | 24 HOUR | 12 | |
| LA | Harbour BBS | 504-673-8092 | C | 24 HOUR | 24 | |
| MA | Channel 1 | 617-354-8873 | C | | 96 | Amiga Conference |
| MA | Amiga Land | 617-399-7876 | C | 24 HOUR | 12 | BBS-PCI |
| MA | Photo Talk | *617-472-8612 | C | | 24 | Amiga echomail ONLY |
| MA | ICOMM | *617-598-5616 | C | 24 HOUR | 24 | New# & Baud, Amiga Sec. |
| MA | Wonderland | *617-665-3796 | C | 24 HOUR | 24 | NEW BAUD Fido Amiga Sec. |
| MA | Toolbox | 617-692-5476 | C | 24 HOUR | 24 | Amiga Section |
| MA | Lost Kingdom | 617-754-3829 | C | | 12 | Amiga Section |
| MA | IDCMP | 617-769-8444 | C | | 12 | |
| MA | MindLink | *617-853-7420 | C | 24 HOUR | 12 | Fido, Amiga Section |
| MA | BCS BBS | 617-862-5779 | C | | 12 | Amiga Section |
| MA | Davy Jones Locker | 617-865-3290 | C | 24 HOUR | 24 | Fido, Amiga Section |
| MA | Cantelope | 617-957-3921 | C | 24 HOUR | 12 | C64, Amiga Section |
| ME | Acorn BBS | *207-865-3004 | C | 24 HOUR | 24 | 120 Mb, 8N1, Amiga Sec. |
| MD | Citadel Cyberpunk | 301-236-0038 | C | | 12 | |
| MD | Rowley | *301-340-0246 | C | | 12 | Amiga Section |
| MD | Amiga Exchange | 301-439-6980 | C | 24 HOUR | 12 | Amiga Magazine New # |
| MD | T.A.C. | 301-445-3777 | C | 24 HOUR | 24 | BBS-PCI, 20 Mb |
| MD | Univ. Maryland 1 | 301-455-3630 | C | 24 HOUR | 12 | Fido, Amiga section |
| MD | Sunrise 80 | 301-666-5703 | C | | 12 | Fido, 7E1 only |
| MD | AMIBBS | 301-666-9109 | C | | 24 | |
| MD | Hotline | *301-683-3944 | C | | 12 | Amiga Section |
| MD | Univ. Marylnd 2 | *301-744-8165 | C | 24 HOUR | 12 | Fido, Amiga Section |
| MD | BiFrost BBS | 301-926-0260 | C | 24 HOUR | 24 | BBS-PCI |
| MD | A.P.E.X. | *301-964-1347 | | | 24 | Back on Line - Amiga |
| MI | Chess Board | *313-255-2456 | C | | 12 | Amiga Section |
| MI | Novi Download | 313-348-4479 | C | 24 HOUR | 12 | 348-4477 Voice 1st |
| MI | PC-Amiga | *313-386-5301 | C | 24 HOUR | 12 | 20Mb |
| MI | Slipped Disk | 313-585-8315 | C | 9PM-9AM | 12 | |
| MI | Excelsior | 313-662-6609 | C | 10PM-8PM | 12 | Weekdays |
| MI | State Street | 313-663-0090 | C | | 12 | |
| MI | Kids Klub | 313-828-3854 | C | 24 HOUR | 12 | 20 Mb |
| MI | Metro Detroit | 313-893-7773 | C | 24 HOUR | 12 | Amiga Section |
| MI | W.C.U.G. | *313-971-2714 | C | | 12 | Amiga Section |
| MI | M Net | 313-994-6333 | C | 24 HOUR | 12 | RBBS, Prompt, Join Amiga |
| MI | Enterprise | 517-372-6037 | C | 8PM-10AM | 12 | |
| MI | Multi Board | 616-335-5119 | C | 24 HOUR | 12 | |
| MI | Metropolitan | 616-342-0076 | C | 24 HOUR | 12 | Amiga Section |
| MI | Proto Vision | 616-343-7721 | C | 24 HOUR | 12 | Citadel, Color |
| MI | SABBS | 616-459-7261 | C | 24 HOUR | 12 | Fido, Amiga Section |
| MI | Inner Sanctum | *906-774-1622 | C | | 12 | Amiga Section |
| MN | Phoenix | 612-459-8095 | C | 24 HOUR | 24 | Hit Space at Connect |
| MN | Impulse | 612-535-0568 | C | 24 HOUR | 24 | Amiga Section |
| MN | Amiga Station | 612-739-2370 | C | 24 HOUR | 12 | 100% Amiga |
| MN | Twin City Express | 612-778-0506 | C | 24 HOUR | 24 | Amiga section |
| MN | Computer Place | 612-869-3246 | C | 6PM-10AM | 24 | 24 Hours on weekends |
| MN | Images @ Twilight | 612-884-7951 | C | 24 HOUR | 12 | Hit Space at Connect |
| MO | MDC RCC | 314-232-6881 | C | 24 HOUR | 24 | McDonnall-Douglas Club |
| MO | Palace BBS | 314-335-4902 | C | 24 HOUR | 12 | Online Games 8N1 |
| MO | Gateway Klasified | 314-521-4527 | C | 24 HOUR | 12 | TBBS, Amiga Section |
| MO | Parthenon | 314-522-1460 | C | | 12 | Fido, Amiga Section |
| MO | Links Corner | 314-772-9345 | C | 7PM-5AM | 12 | C64, Amiga Section |
| MO | CUGSL | 314-878-8816 | C | | 12 | C64, Amiga Section |
| MO | Portal | *314-867-9041 | C | 24 HOUR | 12 | 20 Mb, Amiga Section |
| MO | S.A.I.N.T. BBS | 816-461-2879 | C | 11P-11A | 12 | 150 Mb, 8N1 Citadel Amiga |
| MO | Mebbsie | 816-474-1052 | C | | 12 | Amiga is section 14 |
| NC | Excelerated | 704-541-9909 | C | 24 HOUR | 24 | Amiga section |
| NC | Mountain Soft | *704-685-7887 | C | 24 HOUR | 24 | Store BBS |
| NC | Deep Thought | 919-471-6436 | C | 24 HOUR | 24 | Software Distillery |
| NC | Third Wave | 919-556-7975 | C | 24 HOUR | 12 | $ Membership |
| NC | Missing Link | 919-724-7526 | C | 24 HOUR | 12 | |
| NC | MMS II | 919-772-9176 | C | 24 HOUR | 24 | 2nd line to MMS |
| NC | MMS III | 919-779-5059 | C | 24 HOUR | 12 | 3rd line to MMS |
| NC | MMS | 919-779-6674 | C | 24 HOUR | 24 | TBBS, 100Mb, Amiga sec. |
| NC | Alein | 919-945-2818 | C | 8PM-8AM | 12 | |
| ND | Frozen Banana | 701-746-5932 | C | 24 HOUR | 24 | |
| NE | Wind Dragon | 402-291-8053 | C | 24 HOUR | 24 | Fido, Amiga Section |
| NE | IANR Software Xc | 402-472-5412 | C | 24 HOUR | 12 | Big Amiga section |
| NE | Cinderella | 402-477-0016 | C | 24 HOUR | 12 | Amiga only |
| NE | Intrada | *402-558-4745 | C | 24 HOUR | 96 | Amigal |
| NH | The Tardis | 603-749-1017 | C | 24 HOUR | 24 | 8N1, Ansi Color, 20 Mb |
| NH | Amiga Exchange | 603-749-3038 | C | PM-5AM | 24 | BBS-PCI |
| NJ | Excalibur's Realm | 201-256-0691 | C | 24 HOUR | 12 | BBS-PCI on Amiga |
| NJ | The Power Pack | *201-297-3442 | C | 24 HOUR | 24 | 40 Mb, Amiga Section |
| NJ | Drew U II | 201-377-8193 | C | 24 HOUR | 24 | Amiga Section |
| NJ | Drew U I | 201-377-8245 | C | 24 HOUR | 12 | Amiga Section |
| NJ | Villa Strangiato | *201-386-9150 | C | 24 HOUR | 12 | Amiga Section |
| NJ | Graphics Dump | 201-469-0049 | C | 24 HOUR | 12 | Graphics |
| NJ | Micro Fone | 201-494-3649 | C | 24 HOUR | 12 | Echomail |
| NJ | MANX | 201-542-2793 | C | | 12 | Aztec C Support |
| NJ | JABS BBS | 201-563-9057 | C | 24 HOUR | 12 | JAUG Board |
| NJ | Soup Kitchen | 201-752-2820 | C | 24 HOUR | 12 | "RN Amiga" @ prompt |
| NJ | Amiga Annex | *201-798-1979 | C | | 12 | |
| NJ | AMLINE | 201-864-0121 | C | 8PM-8AM | 12 | Run by store |
| NJ | AGJ | 201-886-8041 | C | 24 HOUR | 24 | Fido, Amiga Section |
| NJ | Dragon's Weyr | 201-992-0834 | C | 24 HOUR | 12 | DragonSoft, Amiga Section |
| NJ | Dragon's Weyr II | 201-992-9891 | C | 24 HOUR | 12 | DragonSoft, Amiga Section |
| NJ | Boardroom | 201-994-5195 | C | 24 HOUR | 12 | Amiga Section |
| NJ | Vival Amigal | 609-227-8278 | C | 24 HOUR | 12 | |
| NJ | K-M-M-M | *609-232-0773 | C | 24 HOUR | 96 | Pascal, Amiga Section |
| NJ | Second Ring | *609-629-3741 | C | | 24 | HAM BBS Amiga |
| NJ | Paranoia Central | 609-768-8174 | C | 24 HOUR | 12 | Amiga Section |
| NM | Albuquerque RCPM | 505-299-5974 | C | 24 HOUR | 24 | RCPM, Amiga section |
| NM | Fast-Kode | 505-437-2280 | C | 24 HOUR | 24 | Amiga section |
| NM | Messila Valley | 505-524-8372 | C | 24 HOUR | 12 | Big RCPM |
| NV | Hubert | 702-322-8877 | C | | 24 | Many Files |
| NV | The Other BBS | *702-438-3625 | C | 24 HOUR | 24 | on Amiga 500 |
| NV | Investment Haven | *702-348-0998 | C | 8PM-7AM | 24 | Amiga Section |
| NV | Computer Addicts | 702-731-3178 | C | 24 HOUR | 12 | C64, Amiga Files |
| NY | AMUSE | 212-269-4879 | C | 24 HOUR | 24 | Fidonet 107/34 |
| NY | Dorsai Embassy 1 | *212-431-1944 | C | 24 HOUR | 12 | 256 Mb, Amiga Section |
| NY | Dorsai Embassy 2 | *212-431-1948 | C | 24 HOUR | 12 | 256 Mb, Amiga Section |
| NY | Jesuit BBS | 212-579-2869 | C | 24 HOUR | 24 | TBBS, Amiga Section |
| NY | Dorsai Embassy 3 | *212-925-8231 | C | 24 HOUR | 24 | 256 Mb, Amiga Section |
| NY | Super 68000 | 212-927-6919 | C | 24 HOUR | 12 | Amiga Section |
| NY | Dorsai Embassy 4 | *212-966-6406 | C | 24 HOUR | 12 | 256 Mb, Amiga Section |
| NY | Dorsai Embassy 5 | *212-966-4653 | C | 24 HOUR | 24 | 256 Mb, Amiga Section |
| NY | ChipHead BBS | *315-492-9071 | C | 8PM-7AM | 12 | Monday - Friday |
| NY | Disaster Area | *315-668-9216 | C | 24 HOUR | 24 | Amiga Section |
| NY | A.M.U.G. | 516-234-6046 | C | 24 HOUR | 24 | AMIGA, BBS-PCI 4.20 |
| NY | Underground | *516-286-4832 | C | 24 HOUR | 12 | Amiga |
| NY | Y.C.A.M.P. | 516-295-0823 | C | 24 HOUR | 12 | Amiga Music Section |
| NY | Lion's Den | *516-399-1928 | C | 24 HOUR | 12 | TAG BBS, Amiga |
| NY | Rivendell | *516-466-4623 | C | 24 HOUR | 12 | BBS-PCI, Amiga |
| NY | Advantage | 516-661-4881 | C | 24 HOUR | 12 | BBS-PCI, Amiga |
| NY | AfterImages SysA | 516-661-9284 | C | 24 HOUR | 24 | Citadel,10 Mb,Amiga Sec. |
| NY | Zeitgeist | 516-689-3105 | C | 24 HOUR | 24 | Users, Developers, Store |
| NY | Computer Palace | 516-698-6182 | C | 24 HOUR | 12 | ST, Amiga Section, Store |
| NY | Land Of Confusion | *516-737-2739 | C | 24 HOUR | 12 | Color, Amiga Section |
| NY | C.U.B. | *516-783-0260 | C | 24 HOUR | 12 | 1st Amiga 500 BBS |
| NY | Network 2000 BBS | *516-867-4469 | C | 24 HOUR | 12 | 20 Mb, BBS-PCI |
| NY | East Coast MIDI | *516-928-4986 | C | 24 HOUR | 24 | MIDI, Amiga Section |
| NY | Brand X Fido | *518-489-8968 | C | 24 HOUR | 24 | Formerly in Conneticut |
| NY | Home Computer | *716-272-0177 | C | 24 HOUR | 12 | 20 Mb |
| NY | 5th Generation | 716-377-3985 | C | 24 HOUR | 12 | |
| NY | Amiga Odessey | *716-394-3585 | C | 24 HOUR | 12 | 22 Mb, 100% Amiga |
| NY | Raven's Nest | 716-833-2852 | C | 24 HOUR | 24 | Amiga section |
| NY | Software Sales | 716-873-5321 | C | 8PM-9AM | 24 | 24 Hours Weekends |
| NY | Amiga Channel | *718-331-3191 | C | | 12 | Amiga |
| NY | Compu-Net II | 718-331-3195 | C | 24 HOUR | 12 | Amiga Section |
| NY | Gateway NRA | 718-338-3501 | C | 5PM-8AM | 24 | 24 Hours Weekends |
| NY | Smurf's Turf | *718-526-5395 | C | 24 HOUR | 12 | Amiga Message Section |
| NY | Brass Tacks | 718-699-2564 | C | 24 HOUR | 24 | Amiga Section |
| NY | Broadway BBS | *718-852-2823 | C | 24 HOUR | 12 | ALL Amigal |
| NY | Computer Discoun | 914-356-2849 | C | 7PM-10A | 12 | Store |
| NY | MUSICNET | 914-442-4006 | C | 24 HOUR | 12 | MIDI Board, $75/YR |
| NY | HudsonVly Amicus | 914-561-0864 | C | 24 HOUR | 12 | Amiga Section |
| NY | ESP BBS | 914-683-8912 | C | 24 HOUR | 12 | Fido, 30 Mb, Amiga Sec. |
| NY | Electronic NY | 914-735-9362 | C | 24 HOUR | 24 | Citadel, Hit space |
| NY | Eyes of K H | 914-779-5886 | C | 24 HOUR | 12 | C128, Amiga section |
| OH | Firehouse | 216-352-9499 | C | | 12 | C64, Amiga Section 7E1 |
| OH | Telecomm Networ | 216-486-0348 | C | 24 HOUR | 24 | Two Nodes |
| OH | CAUG | *216-581-2284 | C | 24 HOUR | 24 | |
| OH | LCE | 216-644-0028 | C | | 12 | |
| OH | North Coast | 216-953-8057 | C | 24 HOUR | 12 | |
| OH | Shareware | 513-531-4654 | C | 24 HOUR | 24 | Amiga Section 80 |
| OH | Imagination | 513-787-3777 | C | | 24 | Amiga Section |
| OH | Pixel Palace | 513-868-7675 | C | | 12 | |
| OH | The Portal | 614-237-0285 | C | 24 HOUR | 24 | IBM All Amiga New Name |
| OH | The Rom Chip | 614-237-8455 | C | 24 HOUR | 24 | 100% Amiga |
| OH | Little Italy | 614-457-6337 | C | 24 HOUR | 24 | 4 MbAmiga |
| OH | MIDI info Exch. | 614-848-5947 | C | 24 HOUR | 12 | Amiga Section |
| OH | Elect. Cronicle | 614-855-1024 | C | | 24 | Amiga Section |
| OH | Earthrise | 614-868-0906 | C | | 12 | |
| OK | 2nd Hand Software | *405-354-1024 | C | 24 HOUR | 24 | Amiga, 50 Mb |
| OK | Frontier | 405-624-2429 | C | 24 HOUR | 24 | Amiga Section |
| OK | Okla City Amiga | 405-631-9040 | C | | 12 | |
| OR | Amiga Board | 503-284-0443 | C | 24 HOUR | 24 | Tag-BBS, Formaly in L.A. |
| OR | NAG BBS | 503-288-1918 | C | | 12 | Amiga Users Group |
| OR | Animagination | *503-342-1298 | C | 12PM-9A | 12 | (ESAUG) |
| OR | NAG BBS II | 503-656-7393 | C | | 12 | Nag 2nd line |
| OR | OrtLand | 503-726-1815 | C | 22 HOUR | 96 | Echomail & Files |
| OR | OrtLand II | 503-746-6990 | C | 24 HOUR | 24 | Second line |
| PA | DELCHUG | 215-363-6625 | C | 24 HOUR | 24 | TBBS, Amiga Section |
| PA | Micro World | 215-461-6352 | C | 24 HOUR | 12 | |
| PA | Philadelphia Amig | 215-533-3191 | C | 24 HOUR | 24 | |
| PA | KeyStone BBS | *215-770-0774 | C | | 12 | Amiga Section |
| PA | PACS-1 | *215-951-1863 | C | 24 HOUR | 12 | Voice 215-951-1255 1st |
| PA | PACS-2 | *215-951-1864 | C | 24 HOUR | 12 | Voice 215-951-1255 1st |
| PA | PACS-3 | *215-951-1865 | C | 24 HOUR | 12 | Voice 215-951-1255 1st |
| PA | PACS-4 | *215-951-1866 | C | 24 HOUR | 12 | Voice 215-951-1255 1st |

| | | | | | | |
|---|---|---|---|---|---|---|
| PA | Onix (Pay Board) | 215-896-9020 | C | 24 HOUR | 12 | 150 Mb, 8 Lines, $$$ |
| PA | Thunderbird | 412-795-3994 | C | | 12 | All Amiga |
| PA | Beacon Hill | *412-962-9514 | C | 24 HOUR | 24 | RBBS, Amiga Section |
| PA | The Library | 717-534-1460 | C | 24 HOUR | 12 | |
| PA | The Tusk | 717-560-1750 | C | 24 HOUR | 12 | BBS-PCI on Amiga 20 Mb |
| PA | Lifesaver | 814-339-6042 | C | | 96 | Amiga Section |
| RI | RI Underground | 401-683-5482 | C | 24 HOUR | 24 | Color, Amiga Section |
| RI | Amiga Forum | 401-732-3286 | C | 24 HOUR | 12 | 20 Mb, Color Menus |
| SC | Galt's Gulch | 803-252-0220 | C | 24 HOUR | 12 | 1.5 MbAmiga, 3 Drives |
| SC | Critical Mass | 803-366-6235 | C | 24 HOUR | 24 | |
| SC | Board Room | 803-548-1243 | C | | 12 | Amiga Section |
| SC | Causers | 803-796-3127 | C | 8PM-9AM | 12 | All Day Sunday |
| TN | Amiga Advocate | 615-228-3159 | C | 24 HOUR | 24 | Fido, All Amiga |
| TN | Abel Supply | 615-453-0643 | C | 24 HOUR | 12 | On line store |
| TN | Comm-BBS | *615-885-4162 | C | | 12 | C64, Amiga Section |
| TN | Duck Pond | 901-755-5330 | C | | 12 | Fido, $ Board |
| TX | Rising Star | 214-231-1372 | C | 24 HOUR | 24 | Fidonet Amiga Section 3 |
| TX | DMUG | 214-276-8902 | C | 24 HOUR | 24 | American Midi Users |
| TX | S.C.O.P.E | 214-288-1537 | C | 24 HOUR | 24 | 8N1, 60 Mb, $ fee |
| TX | Excaliber | 214-341-2775 | C | 24 HOUR | 24 | BBS-PCI Color Menus |
| TX | AAA II BBS | 512-339-7134 | C | 7PM-10A | 12 | $ Membership |
| TX | AAA III BBS | *512-440-1041 | C | 24 HOUR | 12 | 20 Mb, 100% Amiga |
| TX | Starship Amiga | *512-558-9732 | C | | 24 | BBS-PCI, Amiga |
| TX | Star Wars | *512-673-5355 | C | | 12 | BBS-PCI, Amiga |
| TX | Last Precinct | *512-681-9448 | C | | 24 | Amiga Section |
| TX | AMICOM | 512-737-2552 | C | | 24 | HOUR 24 |
| TX | Output BBS | *512-829-5601 | C | | 24 | Amiga Section. |
| TX | TouchStone | *512-834-0263 | C | | 12 | Amiga EchoMail |
| TX | Phydeaux House | *713-482-7285 | C | 8PM-9AM | 12 | Amiga Section |
| TX | Pasadena Conn | 713-487-6715 | C | | 12 | Amiga Section |
| TX | Apple Astronomy | 713-526-5671 | C | 24 HOUR | 24 | Amiga Section |
| TX | Radar's Lil BBS | 713-931-3753 | C | | 24 | |
| TX | Sugarland | 713-933-2440 | C | | 12 | Fido, Amiga Section |
| TX | Info-Line | *713-977-2702 | C | 8PM-9AM | 12 | Amiga Section |
| TX | Track ][ BBS | 817-244-4151 | C | 6PM-9AM | 12 | 24 Hours Sundays/Holidays |
| TX | House of Dave BBS | 817-566-6746 | C | 24 HOUR | 12 | 2 Drives, 1.5 Mb |
| TX | Nomads Nook | 817-926-8922 | C | 24 HOUR | 24 | |
| TX | Iconoclast | 915-833-3899 | C | 24 HOUR | 12 | Skyline BBS (BETA) |
| UT | Far Side | 801-295-8515 | C | | 12 | 30 Mb, Amiga Section |
| UT | Armadillo | 801-484-2766 | C | | 12 | Fido, Run by store |
| UT | Amiga Central | 801-943-2011 | C | | 24 | Amiga Section |
| VA | Empire I/Gateway | 703-352-1936 | C | | 12 | 9PM Sun to 5PM Fri |
| VA | Amiga Central Ex. | 703-378-6294 | C | 6PM-7AM | 12 | 24 Hours Sunday |
| VA | Argen | 703-435-5949 | C | 24 HOUR | 24 | |
| VA | Amiga Solutions | 703-743-4328 | C | 10PM-8A | 12 | |
| VA | Commodore | *800-833-9400 | C | 24 HOUR | 12 | Rates, $9.95 Month + Time |
| VA | Amiga East | 804-499-2266 | C | | 24 | Commlink |
| VA | Nexus | *804-547-1065 | C | 9PM-6AM | 12 | 40 Mb, Amiga |
| VA | Shadetree Softwar | 804-846-1880 | C | | 12 | Fido, Amiga Sec. New name |
| WA | XAKEPOBO MECTO | 206-324-0830 | C | 24 HOUR | 12 | PC Clone,30 Mb,Amiga Sec |
| WA | PD-Silo 2 | 206-347-0243 | C | 24 HOUR | 12 | |
| WA | Eskimo North 5 | *206-361-8753 | C | 24 HOUR | 24 | Alternate Number |
| WA | Eskimo North 6 | *206-361-8759 | C | 24 HOUR | 24 | Alternate Number |
| WA | Eskimo North 3 | *206-365-0641 | C | 24 HOUR | 24 | Alternate Number |
| WA | Eskimo North 4 | *206-365-1724 | C | 24 HOUR | 24 | Alternate Number |
| WA | Eskimo North 8 | *206-365-5458 | C | 24 HOUR | 24 | Alternate Number |
| WA | Eskimo North 1 | *206-367-3837 | C | 24 HOUR | 12 | |
| WA | Eskimo North 2 | *206-367-5458 | C | 24 HOUR | 24 | Alternate Number |
| WA | Eskimo North 7 | *206-387-2379 | C | 24 HOUR | 24 | Alternate Number |
| WA | Amiga-X | 206-459-2487 | C | | 24 | |
| WA | Start Computing | 206-475-2287 | C | | 12 | Mac, Amiga Section |
| WA | PD-Silo 3 | 206-488-4309 | C | 24 HOUR | 12 | |
| WA | The Forum | 206-565-1476 | C | | 12 | |
| WA | Computers + | 206-581-5155 | C | Evenings | 12 | Store BBS |
| WA | Random Access | 206-582-0906 | C | 8PM-6AM | 12 | |
| WA | C.C.U.G. | 206-736-2628 | C | 7PM-9AM | 12 | C64, 8N1, Amiga SectionA |
| WA | N.W. Commodore | 206-743-6021 | C | | 12 | $ Club User Group |
| WA | Questor | *206-771-7758 | C | | 12 | Amiga Section |
| WA | A-Link BBS | 206-774-4735 | C | 24 HOUR | 12 | 40 MB, Amiga, Color |
| WA | PD-Silo 1 | 206-775-2650 | C | 24 HOUR | 12 | Amiga Section |
| WA | Alki Express | 206-937-5800 | C | | 12 | |
| WA | SPSAUG | 206-943-4548 | C | | 24 | |
| WA | InterMountain | 509-529-7229 | C | | 24 | Nights & Weekends Only |
| WI | After Hours BBS | 414-235-9164 | C | 7PM-8AM | 12 | 5PM SAT. TO 8 AM MON. |
| WI | PSA-BBS | 414-278-5390 | C | 24 HOUR | 24 | 20 Mb, BBS-PCI |
| WI | Generic BBS | 414-282-4181 | C | 24 HOUR | 12 | Opus, Amiga Echomail |
| WI | Fax's Exchange | *414-466-4039 | C | 24 HOUR | 24 | Amiga Section |
| WI | Koins | *414-547-6619 | C | 24 HOUR | 24 | Amiga Section |
| WI | AppleGate | 414-738-1219 | C | | 12 | Opus, Amiga Section |
| WI | S.U.E. | 414-762-6475 | C | 24 HOUR | 12 | Multi User $ Board |
| WI | Comm-Link | 414-784-2096 | C | 24 HOUR | 24 | 30 Mb, Amiga Section |
| WI | FDL-BBS | 414-921-8448 | C | Noon-8AM | 12 | ST, Amiga Section |
| WI | Amiga Zone | 414-968-2462 | C | 10P-7AM | 12 | |
| WI | M.O.O.S.I.E. | 608-249-5657 | C | 24 HOUR | 24 | 20 Mb,ST & Amiga |
| WI | MACUG | 608-251-6612 | C | 24 HOUR | 12 | Commodore Users Group |
| WI | MACC | 608-263-6057 | C | 24 HOUR | 24 | RBBS Amiga at prompt |
| WI | Soul Forge | 608-837-0453 | C | 24 HOUR | 12 | ST, Amiga Section |
| WI | Amiga Works | 608-838-4843 | C | 24 HOUR | 12 | |
| WI | Daily Planet | 608-873-4700 | C | 24 HOUR | 12 | |
| WV | BBS-WV | 304-636-9097 | C | 24 HOUR | 24 | 20 Mb, Amiga Section |
| WV | Graphics Connect | *304-768-9360 | C | 24 HOUR | 12 | 20Mb, Amiga Section |

5 Reasons Why You're Ready For

MacroModem

1. You love telecom, but not memorization. MacroModem's user-written macro libraries and companion help screens (36 macros per file) store log on procedures, remote system menus and commands, .....

2. You've always wanted to use the mouse after you're connected, too. Write macros that mimic remote system commands and menus, then execute them with the mouse or keyboard.

3. You like automation, but not script languages. Our macros use normal commands from MacroModem, remote systems, and AmigaDOS, as well as text and control codes. A multi-windowed MacroEditor is included. No new programming language to learn.

4. You want to do other things while downloading a file. MacroModem is truly multi-tasking, with a NewCLI available anytime, even during file transfers. And MacroModem's error checking won't stop downloads unless you tell it to.

5. Of course MacroModem includes standard telecom software features, too. Teach MacroModem what you want, and it will remember for you.

**MacroModem** - the better way to do telecommunications. $69.95

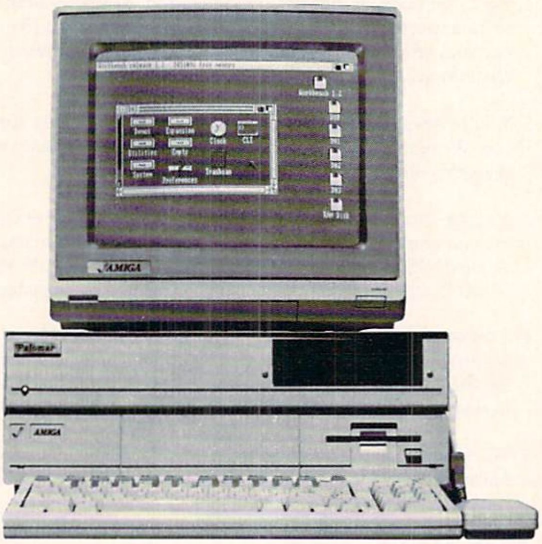
Kent Engineering & Design
P.O. Box 178, Mottville, NY 13119
(315) 685-8237

## CANADA

| | | | | | | |
|---|---|---|---|---|---|---|
| CAN | Unsound Mind | 204-837-9044 | C | | 24 | Winnipeg Amiga |
| CAN | Bit Bucket | 306-352-3236 | C | | 12 | Amiga Section |
| CAN | Calgary Computer | 403-243-2272 | C | | 12 | Amiga Section |
| CAN | Phase 4 | 403-258-0844 | C | 630P-8A | 12 | 24 Hour Sunday, Ram: |
| CAN | Hindson's BBS | *403-328-0320 | C | 6PM-9AM | 12 | 32 Mb, Amiga Section |
| CAN | Edmonton RCPM | 403-454-6093 | C | | 12 | RCPM |
| CAN | C.S.S. | *416-229-4514 | C | | 12 | Evenings & Weekends |
| CAN | C.R.S. | *416-231-0528 | C | 24 HOUR | 12 | PCboard 11.8, Amiga Sec. |
| CAN | Inner Sphere | *416-231-9392 | C | 24 HOUR | 12 | Tag BBS, Amiga Sec. |
| CAN | Amiga Valley BBS | *416-272-4878 | C | 24 HOUR | 12 | BBS-PCI, 45 Mb, |
| CAN | Softor | *416-265-8253 | C | 24 HOUR | 12 | Opus, Amiga Section |
| CAN | AMCIS 1 | 416-427-0268 | C | 24 HOUR | 24 | Fido, Amiga section |
| CAN | O. Science C. | *416-429-1700 | C | 24 HOUR | 12 | Red Ryder |
| CAN | SLC Brampton | *416-452-0408 | C | 24 HOUR | 12 | Opus, Amiga Section |
| CAN | Serial Bowl | *416-453-1081 | C | 4PM-10P | 12 | Commlink, Amiga Only |
| CAN | Millenium II | *416-462-9660 | C | 24 HOUR | 12 | Opus, Amiga Only |
| CAN | Excalibur | *416-484-0607 | C | 24 HOUR | 12 | BBS-PCI, Amiga Section |
| CAN | TROFF | 416-529-6710 | C | 24 HOUR | 24 | Store Board |
| CAN | Track 36 | 416-544-0011 | C | 24 HOUR | 12 | RBBS, 30 Mb, Amiga Sec. |
| CAN | Amiga Intuition | 416-654-8847 | C | 24 HOUR | 12 | BBS-PCI, Amiga |
| CAN | Ziebmef | *416-654-8854 | C | 24 HOUR | 12 | Tag BBS, Amiga Section |
| CAN | WDIX | 416-668-2078 | C | 24 HOUR | 12 | Whitby Info Exchange |
| CAN | BoreDUMB? | *416-694-4702 | C | 24 HOUR | 12 | Opus, Amiga Section |
| CAN | AMCIS 2 | 416-728-9195 | C | 24 HOUR | 12 | Fido, Amiga section |
| CAN | Millenium I | *416-743-9187 | C | 24 HOUR | 12 | Opus, Amiga Section |
| CAN | A-Board | 416-783-7150 | C | 24 HOUR | 12 | BBS-PCI, Amiga |
| CAN | Neutral Zone | *416-897-2594 | C | 24 HOUR | 12 | Amiga & ST |
| CAN | Star Trek | 416-920-6839 | C | 8PM-8AM | 12 | Amiga Only |
| CAN | AMNET | 514-645-3730 | C | | 12 | |
| CAN | Le System Online | 514-844-1567 | C | 24 HOUR | 24 | $35 Year Membership |
| CAN | Montreal | 514-989-1567 | C | 24 HOUR | 24 | |
| CAN | Ottawa Online | 613-526-4141 | C | 24 HOUR | 12 | |
| CAN | OMX | 613-731-3419 | C | | 12 | Color |
| CAN | Connect2 | 705-652-3506 | C | 24 HOUR | 12 | |
| CAN | Norlink BBS | *807-935-6871 | C | | 12 | Amiga Section |

•AC•

**Looking For More "Byte" For Your Dollar?**



Here is the answer, the model **PPI-1000** Expansion Unit for your Amiga 1000 by **Palomar Peripherals.**

☐ Full auto configuration
☐ Meets Zorro specifications
☐ Amiga buss pass-thru
☐ Built in power supply
☐ Power on/off controlled by Amiga
☐ Real time clock with battery backup
☐ Fast RAM
  -2,4,6 or 8 megabytes
  -No wait states
☐ Disk drives
  -Supports up to 2 drives
  -20 or 40 megabyte hard drive(s)
  -3½'' or 5¼'' floppy drive
  -Fast DMA interface

With 20 meg hard drive and 2 meg RAM price is $1395 plus shipping and handling (Calif. Res. add 6% sales tax). Requires Amiga DOS 1.2. For further information call **(619) 748-1202** or write to **Palomar Peripherals.**

**PALOMAR PERIPHERALS**
**14580 High Pine**
**Poway, Ca 92064**

D - Five Associates
19 Crosby Drive
Bedford, MA
01730-0523
(617) 275-8892

Tired of the high cost of computer repairs?
→ FLAT Labor charges
→ FREE Estimates
→ Warranty work

Also:
1764 to 512K: $^{\$}61^{95}$
128 64K vdc RAM: $^{\$}40^{00}$
NEW: C=1902 conversion to RGB-I: $^{\$}40^{00}$

$19.95* C=64/128 & peripherals
$34.95* Commodore PC-10
29.95* Amiga 1000
* plus parts and sales tax

Authorized Commodore Service Center

## Index of Advertisers

A-Squared Distributions Inc. 1
Central Coast Software 27
COMMAND Simulations 91
Comp-U-Save 53
Computer Mart 94
Conflict Recreations 98
CPE Systems 93
D-Five Associates 104
Datamax Research Corp. 78
Delphi Noetic Systems, Inc. 69
Fuller Computer Systems, Inc. 76
Geodesic Publications 73
Haitex Resources 15
Hilton Android 14
HyperTek/Silicon Springs 37
Kent Engineering & Design 103
LampLighter Software, Inc. 75
Lattice, Inc. 7
Lightning Publishing 30
Lynn's Luna C 79
Megatronics CII
MicroBotics, Inc. 11
Microillusions CIII
Microsmiths, Inc. 40
NewTek CIV
Newwave Software 99
Palomar Peripherals 104
Peacock Systems, Inc. 28
Pioneer Computing 54
Prolific Inc. 3
PVS Publishing 24
Rittinghouse Software Development Co. 77
Sedona Software 33
Software Advantage Consulting Corp. 80
Software Integration Solutions 52
Spirit Technology 74
Syndesis 72
The Memory Location 88
The Other Guys 23
The Right Answers Group 13
TRU-IMAGE 3
William S. Hawes 92

# The AMICUS & Fred Fish Public Domain Software Library

This software is collected from user groups and electronic bulletin boards around the nation. Each Amicus disk is nearly full, and is fully accessible from the Workbench. If source code is provided for any program, then the executable version is also present. This means that you don't need the C compiler to run these programs. An exception is granted for those programs only of use to people who own a C compiler.

The Fred Fish disk are collected by Mr. Fred Fish, a good and active friend of the Amiga.

Note: Each description line below may include something like 'S-O-E-D', which stands for 'source, object file, executable and documentation'. Any combination of these letters indicates what forms of the program are present. Basic programs are presented entirely in source code format.

## AMICUS Disk 1

**ABasic programs: Graphics**

| | |
|---|---|
| 3DSolids | 3d solids modeling prog. w/sample data files |
| Blocks | draws blocks |
| Cubes | draws cubes |
| Durer | draws pictures in the style of Durer |
| FScape | draws fractal landscapes |
| Hidden | 3D drawing program, w/ hidden line removal |
| JPad | simple paint program |
| Optical | draw several optical illusions |
| PaintBox | simple paint program |
| Shuttle | draws the Shuttle in 3d wireframe |
| SpaceArt | graphics demo |
| Speaker | speech utility |
| Sphere | draws spheres |
| Spiral | draws color spirals |
| ThreeDee | 3d function plots |
| Topography | artificial topography |
| Wheels | draws circle graphics |
| Xenos | draws fractal planet landscapes |

**ABasic programs: Tools**

| | |
|---|---|
| AddressBook | simple database program for addresses |
| CardFile | simple card file database program |
| Demo | multiwindow demo |
| KeyCodes | shows keycodes for a key you press |
| Menu | run many ABasic programs from a menu |
| MoreColors | way to get more colors on the screen at once, using aliasing |
| shapes | simple color shape designer SpeakIt speech and narrator demo |

**ABasic programs: Games**

| | |
|---|---|
| BrickOut | classic computer brick wall game |
| Othello | also known as 'go' |
| Saucer | simple shoot-em-up game |
| Spelling | simple talking spelling game |
| ToyBox | selectable graphics demo |

**ABasic programs: Sounds**

| | |
|---|---|
| Entertainer | plays that tune |
| HAL9000 | pretends it's a real computer |
| Police | simple police siren sound |
| SugarPlum | plays "The Dance of the Sugarplum Fairies" |

**C programs:**

| | |
|---|---|
| ATerm | simple terminal program, S-E |
| cc | aid to compiling with Lattice C |
| decvnt | opposite of CONVERT for cross developers |
| Dotty | source code to the 'dotty window demo |
| echox | unix-style filename expansion, partial S,O-D |
| fasterfp | explains use of fast-floating point math |
| FixDate | fixes future dates on all files on a disk, S-E |
| freedraw | simple Workbench drawing prog., S-E |
| GfxMem | graphic memory usage indicator, S-E |
| Grep | searches for a given string in a file with |
| docs. | ham shows off the hold-and-modify method of color generation |
| IBM2Amiga | fast parallel cable transfers between an IBM and an Amiga |
| Mandel | Mandelbrot set program, S-E |
| moire | patterned graphic demo, S-E |
| objfix | makes Lattice C object file symbols visible to Wack, S-E |
| quick | quick sort strings routine |
| raw | example sample window I/O |
| setlace | turns on interlace mode, S-E |
| sparks | qix-type graphic demo, S-E |

**Other executable programs:**

| | |
|---|---|
| SpeechToy | speech demonstration |
| WhichFont | displays all available fonts |

**Texts:**

| | |
|---|---|
| 68020 | describes 68020 speedup board from CSA |
| Aliases | explains uses of the ASSIGN command |
| Bugs | known bug list in Lattice C 3.02 |
| CLICard | reference card for AmigaDOS CLI |
| CLICommands | guide to using the CLI |
| Commands | shorter guide to AmigaDOS CLI commands |
| EdCommands | guide to the ED editor |
| Filenames | AmigaDOS filename wildcard conventions |
| HalfBright | explains rare graphics chips that can do more colors |
| ModemPins | description of the serial port pinout |
| RAMdisks | tips on setting up your RAM: disk |
| ROMWack | tips on using ROMWack |
| Sounds | explanation of Instrument demo sound file format |
| Speed | refutation of Amiga's CPU and custom chip speed |
| WackCmds | tips on using Wack |

## AMICUS Disk 2

**C programs:**

| | |
|---|---|
| alib | AmigaDOS object library manager , S-E |
| ar | text file archive program, S-E |
| fixobj | auto-chops executable files |
| shell | simple CLI shell, S-E |
| sq, usq | file compression programs, S-E |
| YachtC | a familiar game, S-E |
| Make | a simple 'make' programming utility, S-E |
| Emacs | an early version of the Amiga text editor, S-E-D |

**Assembler programs:**

| | |
|---|---|
| bsearch.asm | binary search code |
| qsort.asm | Unix compatible qsort() function, source and C test program |
| setjmp.asm | setjmp() code for Lattice 3.02 |
| SVprintf | Unix system V compatible printf() |
| trees.o | Unix compatible tree() function, O-D |

(This disk formerly had IFF specification files and examples. Since this spec is constantly updated, the IFF spec files have been moved to their own disk in the AMICUS collection.)

**John Draper Amiga Tutorials:**

| | |
|---|---|
| Animate | describes animation algorithms |
| Gadgets | tutorial on gadgets |
| Menus | learn about Intuition menus |

## AMICUS Disk 3

**C programs:**

| | |
|---|---|
| Xref | a C cross-reference gen., S-E |
| 6bitcolor | extra-half-bright chip gfx demo, S-E |
| Chop | truncate (chop) files down to size, S-E |
| Cleanup | removes strange characters from text files |
| CR2LF | converts carriage returns to line feeds in Amiga files, S-E |
| Error | adds compile errors to a C file, S |
| Hello | window ex. from the RKM, S |
| Kermit | generic Kermit implementation, flakey, no terminal mode, S-E |
| Scales | sound demo plays scales, S-E |
| SkewB | Rubik cube demo in hi-res colors, S-E |

**AmigaBasicProgs(dir)**

| | |
|---|---|
| Automata | cellular automata simulation |
| CrazyEights | card game |
| Graph | function graphing programs |
| WitchingHour | a game |

**ABasiC programs:**

| | |
|---|---|
| Casino | games of poker, blackjack, dice, and craps |
| Gomoku | also known as 'othello' |
| Sabotage | sort of an adventure game |

**Executable programs:**

| | |
|---|---|
| Disassem | a 68000 disassembler, E-D |
| DpSlide | shows a given set of IFF pictures, E-D |
| Arrange | a text formatting program, E-D |

**Assembler programs:**

| | |
|---|---|
| Argoterm | terminal program with speech and Xmodem, S-E |

## AMICUS Disk 4 Files from the original Amiga Technical BBS

Note that some of these files are old, and refer to older versions of the operating system. These files came from the Sun system that served as Amiga technical support HQ for most of 1985. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

Complete and nearly up-to-date C source to 'image.ed', an early version of the Icon Editor. This is a little flaky, but compiles and runs.

An Intuition demo, in full C source, including files: demomenu.c, demomenu2.c, demoreq.c, getascii.c, idemo.c, idemo.guide, idemo.make, idemoall.h, nodos.c, and txwrite.c

| | |
|---|---|
| addmem.c | add external memory to the system |
| bobtest.c | example of BOB use |
| consoleIO.c | console IO example |
| creaport.c | create and delete ports |
| creastdi.c | create standard I/O requests |
| creatask.c | creating task examples |
| diskio.c | example of track read and write |
| dotty.c | source to the 'dotty window' demo |
| dualplay.c | dual playfield example |
| flood.c | flood fill example |
| freemap.c | old version of 'freemap' |
| geltools.c | tools for VSprites and BOBs |
| gfxmem.c | graphic memory usage indicator |
| hello.c | window example from RKM |
| inputdev.c | adding an input handler to the input stream |
| joystik.c | reading the joystick |
| keybd.c | direct keyboard reading |
| layertes.c | layers examples |
| mousport.c | test mouse port |
| ownlib.c, | |
| ownlib.asm | example of making your own library with Lattice |
| paratest.c | tests parallel port commands |
| seritest.c | tests serial port commands |
| serisamp.c | example of serial port use |
| prinintr.c | sample printer interface code |
| prtbase.h | printer device definitions |
| regintes.c | region test program |
| setlace.c | source to interlace on/off program |
| setparallel.c | set the attributes of the parallel port |
| SetSerial.c | set the attributes (parity, databits)of the |
| singplay.c | single playfield example |
| speechtoy.c | source to narrator and phonetics demo |
| timedely.c | simple timer demo |
| timer.c | exec support timer functions |
| timrstuf.c | more exec support timer functions |
| WhichFont.c | loads and displays all available system fonts |

process.i and prtbase.i assmebler include files:

| | |
|---|---|
| autorqstr.txt | warnings of deadlocks with autorequesters |
| consoleIO.txt | copy of the RKM console I/O chapter |
| diskfont.txt | warning of disk font loading bug |
| fullfunc.txt | list of #defines, macros, functions |
| inputdev.txt | preliminary copy of the input device chapter |

License information on Workbench distribution license

printer pre-release copy of the chapter on printer drivers, from RKM 1.1 v11fd.txt 'diff' of .fd file changes from version 1.0 to 1.1

v28v1.diff 'diff' of include file changes from version 28 to 1.0

## AMICUS Disk 5 Files from the Amiga Link / Amiga Information Network

Note that some of these files are old, and refer to older versions of the operating system. These files are from Amiga Link. For a time, Commodore supported Amiga Link, aka AIN, for online developer technical support. It was only up and running for several weeks. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

**A demo of Intuition menus called 'menudemo', in C source**

| | |
|---|---|
| whereis.c | find a file searching all subdirectories |
| bobtest.c | BOB programming example |
| sweep.c | sound synthesis example |

**Assembler files:**

| | |
|---|---|
| mydev.asm | sample device driver |
| mylib.asm | sample library example |
| mylib.i | |
| mydev.i | |
| asmsupp.i | |
| macros.i | assembler include files |

**Texts:**

| | |
|---|---|
| amigatricks | tips on CLI commands |
| extdisk | external disk specification |
| gameport | game port spec |
| parallel | parallel port spec |
| serial | serial port spec |
| v1.1update | list of new features in version 1.1 |
| v1.1h.txt | 'diff' of include file changes from version |

Files for building your own printer drivers, including dospecial.c, epsondata.c, init.asm, printer.c, printer.link, printertag.asm, render.c, and wait.asm. This disk does contain a number of files describing the IFF specification. These are not the latest and greatest files, but remain here for historical purposes. They include text files and C source examples. The latest IFF spec is elsewhere in this library.

## AMICUS Disk 6 IFF Pictures

This disk includes the DPSlide program, which can view a given series of IFF pictures, and the 'showpic' program, which can view each file at the click of an icon. The pictures include a screen from ArticFox, a Degas dancer, the guys at Electronic Arts, a gorilla, horses, King Tut, a lighthouse, a screen from Marble Madness, the Bugs Bunny Martian, a still from an old movie, the Dire Straits moving company, a screen from Pinball Contruction Set, a TV newcaster, the PaintCan, a world map, a Porsche, a shuttle mission patch, a tyrannosaurus rex, a planet view, a VISA card, and a ten-speed.

## AMICUS Disk 7 DigiView HAM demo picture disk

This disk has pictures from the DigiView hold-and-modify video digitizer. It includes the ladies with pencils and lollypops, the young girl, the bulldozer, the horse and buggy, the Byte cover, the dictionary page, the robot and Robert. This includes a program to view each picture separately, and all together as separate, slidable screens. The 'seeilbm' program, to turn any screen into an IFF picture.

## AMICUS Disk 8

**C programs:**

| | |
|---|---|
| Browse | view text files on a disk, using menus S-E-D |
| Crunch | removes comments and white space from C files, S-E |
| IconExec | EXECUTE a series of commands from Workbench S-E |
| PDScreen Dump | dumps Rastport of highest screen to printer |
| SetAlternate | sets a second image for an icon, when clicked once S-E |
| SetWindow | makes windows for a CLI program to run under Workbench S-E |
| SmallClock | a small digital clockin a window menu bar |
| Scrimper | the screen printer in the fourth AC S-E |

**Amiga Basic Programs:**

(**Note**: Many of these programs are present on AMICUS Disk 1. Several of these were converted to Amiga Basic, and are included here.)

| | |
|---|---|
| AddressBook | a simple address book database |
| Ball | draws a ball |
| Cload | program to convert Compuserve hex files to binary, S-D |
| Clue | the game, Intuition driven |
| ColorArt | art drawing program |
| DeluxeDraw | the drawing program in the 3rd AC, S-D |
| Eliza | conversational computer psychologist |
| Othello | the game, as known as 'go' |
| RatMaze | 3D ratmaze game |
| ROR | boggling graphics demo |
| Shuttle | draws 3D pictures of the space shuttle |
| Spelling | simple spelling program |
| YoYo | wierd zero-gravity yo-yo demo, tracks yo-yo to the mouse |

**Executable programs:**

| | |
|---|---|
| 3Dcube | Modula-2 demo of a rotating cube |
| AltIcon | sets a second icon image, displayed when the icon is clicked |
| AmigaSpell | a slow but simple spell checker, E-D |
| arc | the ARC file compression progra must-have for telecom, E-D |
| Bertrand | graphics demo |
| disksalvage | prog. to rescue trashed disks, E-D |
| KwikCopy | a quick but nasty disk copy program: ignores errors, E-D |
| LibDir | lists hunks in an object file E-D |
| SaveILBM | saves any screen as IFF pic.E-D ?? |
| ScreenDump | shareware screen dump prog, E only |
| StarTerm | version 2.0, term program, XmodemE-D |

**Texts:**

| | |
|---|---|
| LatticeMain | tips on fixing _main.c in Lattice |
| GDiskDrive | make your own 5 1/4 drive |
| GuruMed | explains the Guru numbers |
| Lat3.03bugs | bug list of Lattice C version 3.03 |
| MForgeRev | user's view of the MicroForge HD |
| PrintSpooler | EXECUTE-based print spool prog. |

**.BMAP files:**

These are the necessary links between Amiga Basic and the system libraries. To take advantage of the Amiga's capabilities in Basic, you need these files. BMAPs are included for 'dist', 'console', 'diskfont', 'exec', 'icon', 'intuition', 'layers', 'mathffp', mathieeedoubas', 'mathieeesingbas', 'mathtrans', 'potgo', 'timer' and 'translator'.

## AMICUS Disk 9

**Amiga Basic Programs:**

| | |
|---|---|
| FlightSim | simple flight simulator program |
| HuePalette | explains Hue, Saturation, & Intensity |
| Requester | ex. of requesters from Amiga Basic |
| ScrollDemo | demonstrates scrolling capabilities |
| Synthesizer | sound program |
| WorldMap | draws a map of the world |

**Executable programs:**

| | |
|---|---|
| Boing! | latest Boing! demo,with selectable speed,E |
| Brush2C | converts an IFF brush to C data instructions, initialization code, E |
| Brush2Icon | converts IFF brush to an icon, E |
| Dazzle | graphics demo, tracks to mouse, E |
| DeciGEL | assembler program for stopping 68010 errors, S-E-D |
| Klock | menu-bar clock and date display, E |
| life | the game of life, E |
| TimeSet | Intuition-based way to set the time & date |
| EMEmacs | another Emacs, more oriented to word processing, S-E-D |
| MyCLI | a CLI shell, works without the Workbench, S-E-D |

**Texts:**

| | |
|---|---|
| FnctnKeys | read function keys from Amiga Basic |
| HackerSin | explains how to win the game 'hacker' |
| Ist68010 | guide to installing a 68010 in your Amiga |
| Boing! | latest Boing! demo,with selectablespeed, E |
| Brush2C | converts an IFF brush to C data instructions, initialization code, E |
| Brush2Icon | converts IFF brush to an icon, E |
| Dazzle | graphics demo, tracks to mouse, E |
| DeciGEL | assembler program for stopping 68010 errors, S-E-D |
| Klock | menu-bar clock and date display, E |
| life | the game of life, E |
| TimeSet | Intuition-based way to set the time date, |
| EMEmacs | another Emacs, more oriented to word processing, S-E-D |
| MyCLI | a CLI shell, works without the Workbench, S-E-D |

For PDS orders, Please use form on page 112

**Texts:**

| | |
|---|---|
| FnctnKeys | explains how to read function keys from Amiga Basic |
| HackerSln | explains how to win the game 'hacker' |
| Ist68010 | guide to installing a 68010 in your Amiga |
| PrinterTip | sending escape sequences to your printer |
| StartupTip | tips on setting up your startup-sequence file |
| XfmrReview | list of Transformer programs that work |

**Printer Drivers:**

Printer drivers for the Canon PJ-1080A, the C Itoh Prowriter, an improved Epson driver that eliminates streaking, the Epson LQ-800, the Gemini Star-10, the NEC 8025A, the Okidata ML-92, the Panasonic KX-P10xx family, and the Smith-Corona D300, with a document describing the installation process.

### AMICUS Disk 10 **Instrument sound demos**

This is an icon-driven demo, circulated to many dealers. It includes the sounds of an acoustic guitar, an alarm, a banjo, a bass guitar, a boink, a calliope, a car horn, claves, water drip, electric guitar, a flute, a harp arpegio, a kickdrum, a marimba, a organ minor chord, people talking, pigs, a pipe organ, a Rhodes piano, a saxophone, a sitar, a snare drum, a steel drum, bells, a vibrophone, a violin, a wailing guitar, a horse whinny, and a whistle.

### AMICUS Disk 11

C programs

| | |
|---|---|
| dirutil | Intuition-based, CLI replacement manager S-E |
| cpri | shows and adjusts priority of CLI processes, S-E |
| ps | shows info on CLI processes, S-E |
| vidtex | displays Compuserve RLE pics, S-E |

AmigaBasic programs

| | |
|---|---|
| pointered | pointer and sprite editor program |
| optimize | optimization ex ample from AC article |
| calendar | large, animated calendar, diary and date book program |
| amortize | loan amortizations |
| brushtoBOB | converts small IFF brushes to AmigaBasic BOB OBJECTS |
| grids | draw and play waveforms |
| hilbert | draws Hilbert curves |
| madlib | mad lib story generator |
| mailtalk | talking mailing list program |
| meadows3D | 3D graphics program, from A C™ article |
| mousetrack | mouse tracking example in hires mode |
| slot | slot machine game |
| tictactoe | the game |
| switch | pachinko-like game |
| weird | makes strange sounds |

Executable programs

| | |
|---|---|
| cp | unix-like copy command, E |
| cls | screen clear, S-E |
| diff | unix-like stream editor uses 'diff' output to fix files |
| pm | chart recorder performances indicator |

Assembler programs

| | |
|---|---|
| cls | screen clear and CLI arguments example |

Modula-2

| | |
|---|---|
| trails | moving-worm graphics demo |
| caseconvert | converts Modula-2 keywords to uppercase |
| Forth | Breshehan circle algorithm example |
| Analyze | 12 templates for the spreadsheet Analyze |

There are four programs here that read Commodore 64 picture files. They can translate Koala Pad, Doodle, Print Shop and News Room graphics to IFF format. Getting the files from your C-64 to your Amiga is the hard part.

### AMICUS Disk 12

Executable programs

| | |
|---|---|
| blink | 'alink' compatible linker, but faster, E-D |
| clean | spins the disk for disk cleaners, E-D |
| epsonset | sends Epson settings to PAR from menu E-D |
| showbig | view hi-res pics in low-res superbitmap, E-D |
| speaktime | tell the time, E-D |
| undelete | undeletes a file, E-D |
| cnvapichm | converts Apple ][ low, medium and high res pictures to IFF, E-D |
| menued | menu editor produces C code for menus, E-D |
| quick | quick disk-to-disk nibble copier, E-D |
| quickEA | copies Electronic Arts disks, removes protection, E-D |
| txed 1.3 | demo of text editor from Microsmiths,E-D |

C programs

| | |
|---|---|
| spin3 | rotating blocks graphics demo, S-E-D |
| popcli | start a new CLI at the press of a button, like Sidekick, S-E-D |
| vsprite | VSprite example code from Commodore, S-E-D |
| AmigaBBS | Amiga Basic bulletin board prog., S-D |

**Assembler programs**

| | |
|---|---|
| star10 | makes star fields like Star Trek intro,S-E-D |

Pictures

| | |
|---|---|
| Mount Mandelbrot | 3D view of Mandelbrot set |
| Star Destroyer | hi-res Star Wars starship |
| Robot | robot arm grabbing a cylinder |

Texts

| | |
|---|---|
| vendors | Amiga vendors, names, addresses |
| cardco | fixes to early Cardco memory boards |
| cinclude | cross-reference to C include files |
| mindwalker | clues to playing the game well |
| slideshow | make your own slideshows from the Kaleidoscope disk |

### AMICUS Disk 13

Amiga Basic programs

Routines from Carolyn Scheppner of CBM Tech Support, to read and display IFF pictures from Amiga Basic. With documentation. Also included is a program to do screen prints in Amiga Basic, and the newest BMAP files, with a corrected ConvertFD program. With example pictures, and the SaveILBM screen capture program.

Routines to load and play FutureSound and IFF sound files from Amiga Basic, by John Foust for Applied Visions. With documentation and C and assembler source for writing your own libraries, and interfacing C to assembler in libraries. With example sound.

Executable programs

| | |
|---|---|
| gravity | Sci Amer Jan 86 gravitation graphic simulation, S-E-D |

Texts

| | |
|---|---|
| MIDI | make your own MIDI instrument interface, with documentation and a hi-res schematic picture. |

### AMICUS Disk 14

Several programs from Amazing Computing issues:

Tools

| | |
|---|---|
| Dan Kary's | C structure index program, S-E-D |

Amiga Basic programs:

| | |
|---|---|
| BMAP Reader | by Tim Jones |
| IFFBrush2BOB | by Mike Swinger |
| AutoRequester | example |
| DOSHelper | Windowed help system for CLI commands, S-E-D |
| PETrans | translates PET ASCII files to ASCII files, S-E-D |
| C Squared | Graphics program from Scientific American, Sept 86, S-E-D |
| crlf | adds or removes carriage returns from files, S-E-D |
| dpdecode | decrypts Deluxe Paint, remo |
| ves copy | protection, E-D |
| queryWB | asks Yes or No from the user returns exit code, S-E |
| vc | VisiCalc type spreadsheet, no mouse control, E-D |
| view | views text files with window and |
| slider | gadget, E-D |

Oing, Sproing, yaBoing, Zoing are sprite-based

| | |
|---|---|
| Boing! style | demos, S-E-D |

CLIClock, sClock, wClock are window border clocks, S-E-D

**Texts**

An article on long-persistance phospor monitors, tips on making brushes of odd shapes in Deluxe Paint, and recommendations on icon interfaces from Commodore-Amiga.

### AMICUS Disk 15

**The C programs include:**

| | |
|---|---|
| 'pr' | a file printing utility, which can print files in the background, and with line numbers and control character filtering. |
| 'fm' | displays a chart of the blocks allocated on a disk. |
| 'Ask' | questions an 'execute' file, returns an error code to control the execution in that batch file |
| 'Stat' | an enhanced version of AmigaDOS 'status' command. |
| 'Dissolve' | random-dot dissolve demo displays IFF picture slowly, dot by dot, in a random fashion. |
| 'PopCLI2' | invoke new CLI window at the press of a key. |

**The executable programs include:**

| | |
|---|---|
| 'Form' | file formatting program through the printer driver to select print styles |
| 'DiskCat' | catalogs disks, maintains, sorts,merges lists of disk files |
| 'PSound' | SunRize Industries' sampled sound editor & recorder |
| 'Iconmaker' | makes icons for most programs |
| 'Fractals' | draws great fractal seascapes and mountain scapes. |
| '3D Breakout' | 3D glasses, create breakout in a new dimension |
| 'AmigaMonitor' | displays lists of open files, |
| memory use, | tasks, devices and ports in use. |
| 'Cosmoroids' | version of 'asteroids' for the Amiga. |
| 'Sizzlers' | high resolution graphics demo written in Modula 2. |

**Texts:**

| | |
|---|---|
| 'ansi.txt' | explains escape sequences the CON: device responds to. |
| 'FKey' | includes template for making paper to sit in the tray at the top of the Amiga keyboard. |
| 'Spawn' | programmer's document from Commodore |

Amiga, describs ways to use the Amiga's multitasking capabilities in your own programs.

**AmigaBasic programs:**

| | |
|---|---|
| 'Grids' | draw sound waveforms, and hear them played. |
| 'Light' | a version of the Tron light-cycle video game. |
| 'MigaSol' | a game of solitaire. |
| 'Stats' | program to calculate batting averages |
| 'Money' | "try to grab all the bags of money that you can." |

AMICUS 15 also includes two beautiful IFF pictures, of the enemy walkers from the ice planet in Star Wars, and a picture of a cheetah.

### AMICUS Disk 16

'juggler' demo by Eric Graham. a robot juggler bouncing three mirrored balls, with sound effects. Twenty-four frames of HAM animation are flipped quickly to produce this image. You control the speed of the juggling. The author's documentation hints that this program might someday be available as a product.

**IFF pictures**

parodies of the covers of Amiga World and Amazing Computing magazines.

**C programs:**

| | |
|---|---|
| 'Inputhandler' | example of making an input handler. |
| 'FileZap3' | binary file editing program |
| 'ShowPrint' | displays IFF picture, and prints it. |
| 'Gen' | program indexes and retrieves C structures and variables declared in the Amiga include file system. |

**Executable Programs:**

| | |
|---|---|
| 'FixHunk2' | repairs an executable program file for expanded memory |
| 'ms2smus' | converts Music Studio files to IFF standard 'SMUS' format. I have heard this program might have a few bugs, especially in regards to very long songs, but it works in most cases. |
| 'Missile' | Amiga version of the 'Missile Command' video game, |

This disk also contains several files of scenarios for Amiga Flight Simulator II. By putting one of these seven files on a blank disk, and inserting it in the drive after performing a special command in this game, a number of interesting locations are preset into the Flight Simulator program. For example, one scenario places your plane on Alcatraz, while another puts you in Central Park

### AMICUS Disk 17

Telcommunications disk whuch contains six terminal programs.

| | |
|---|---|
| "Comm" V1.33 | term prog. with Xmodem, WXmodem, |
| "ATerm" V7.2 | term prog. includes Super Kermit |
| "VT-100"V2.6 | Dave Wecker's VT-100 emulator with Xmodem,Kermit, and scripting |
| "Amiga Kermit" | V4D(060) port of the Unix C-Kermit |
| "VTek" V2.3.1 | Tektronix graphics terminal emulator based on the VT-100 prog. V2.3 and contains latest 'arc' file compression |
| "AmigaHost" | V0.9 for Compuserve. Includes RLE graphics abilities & CIS-B file transfer protocol. |
| "FixHunk" | expansion memory necessity |
| "FixObj" | removes garbage characters from modem received files |
| "Txt" | filters text files from other systems to be read by the Amiga E.C. |
| "addmem" | executeable version for use with mem expansion article in AC v2..1 |
| 'arc' | file documentation and a basic tutorial on un 'arcing files |
| "arcre" | for makeing "arc" files E.C. |

### AMICUS Disk 18

| | |
|---|---|
| Logo | Amiga version of the populacomputer language, with example programs, E-D |
| Tv*Text | Demo version of the TV*Text character generator |
| PageSetter | Freely distributable versions of the updated PagePrint and PageIFF programs for the PageSetter desktop publishing package. |
| FullWindow | Resizes any CLI window using only CLI commands, E-D |
| Life3d | 3-D version of Conway's LIFE program, E-D |
| Defdisk | CLI utility to re-assign a new Workbench disk, S-E-D |
| Calendar.WKS | Lotus-compatible worksheet that makes calendars |
| SetKey | Demo of keyboard key re-programmer, with IFF picture to make function key labels, E-D |
| VPG | Video pattern generator for aligning monitors, E-D |
| HP-10C | Hewlett-Packard-like calculator, E-D |
| SetPrefs | Change the Preferences settings on the fly, in C, S-E-D |
| StarProbe | Program studies stellar evolution. C source included for Amiga and MS-DOS, S-E-D |
| ROT | C version of Colin French's AmigaBasic ROT program from Amazing Computing. ROT edits and displays polygons to create three dimensional objects. Up to 24 frames of animation can be created and displayed. E-D |
| Scat | Like Ing, windows on screen run away from the mouse, E-D |
| DK | Decays" the CLI window into dust, in Modula 2, S-E-D |
| DropShadow2 | Adds layered shadows to Workbench windows, E-D |

### AMICUS Disk 19

This disk carries several programs from Amazing Computing. The IFF pictures on this disk include the Amiga Wake part T-shirt logo, a sixteen-color hi-res image of Andy Griffith, and five Amiga Live! pictures from the Amazing Stories episode that featured the Amiga.

| | |
|---|---|
| Solve | Linear equation solver in assembly language, S-E-D |
| Gadgets | Bryan Catley's AmigaBasic tutorial, |
| Household | Bryan Catley's AmigaBasic household inventory program, S-D |
| Waveform | Jim Shields' Waveform WoAmigaBasic, S-D |
| DiskLib | John Kennan's AmigaBasic disk librarian program, S-D |
| Subscripts | Ivan Smith's AmigaBasic subscript example, S-D |
| String, Boolean | C programs and executables for Harriet Maybeck Tolly's Intuition tutorials, S-E-D |
| Skinny C | Bob Riemersma's example for making small C programs, S-E-D |
| COMAL.h | Make C look like COMAL Header file, |
| EmacsKey | Makes Emacs function key definitons by Greg Douglas, S-D |
| AMon 1.1 | Snoop on system resource use, E-D |
| BTE | Bard's Tale character editor, E-D |
| Size | CLI program shows the size of a given set of files, E-D |
| WinSize | CLI window utility resizes current window, S-E-D |

### AMICUS Disk 20

Compactor, Decoder Steve Michel AmigaBasic tools, S-D

| | |
|---|---|
| BobEd | BOB and sprite editor written in C,S-E-D |
| SpriteMasterII | Sprite editor and animator by Brad Kiefer, E-D |
| BlitLab | Blitter chip exploration C program by Tomas Rokicki, S-E-D |
| FPic | Image processing program by Bob Bush loads and saves IFF images, changes them with several techniques, E-D |
| Bankn | Complete home banking program, balance your checkbook! E-D |
| cons | Console device demo program with supporting macro routines. |
| freemap | Creates a visual diagram of free memory |
| input.dev | sample input handler, traps key or mouse events |
| joystick | Shows how to set up the gameport device as a joystick. |
| keyboard | demonstrates direct communications with the keyboard. |
| layers | Shows use of the layers library |
| mandelbrot | IFF Mandelbrot program |
| mouse | hooks up mouse to right joystick port |
| one.window | console window demo |
| parallel | Demonstrates access to the parallel port. |
| printer | opening and using the printer, does a screen dump, not working |
| print.support | Printer support routines, not working. |
| proctest | sample process creation code, not working |
| region | demos split drawing regions |
| samplefont | sample font with info on creating your own |
| serial | Demos the serial port |
| singlePlayfield | Creates 320 x 200 playfield |
| speechtoy | latest version of cute speech demo |
| speech.demo | simplified version of speechtoy, with IO requests |
| text.demo | displays available fonts |
| timer | demos timer.device use |
| trackdisk | demos trakcdisk driver |

### AMICUS Disk 21

| | |
|---|---|
| Target | Makes each mouse click sound like a gunshot, S-E-D |
| Sand | Simple game of sand that follows the mouse pointer, E-D |
| PropGadget | Harriet Maybeck Tolly's proportional gadget example, S-E |
| EHB | Checks to see if you have extra-half-bright graphics, S-E-D |
| Piano | Simple piano sound program |
| CelScripts | Makes cel animation scripts for Aegis Animator, in AmigaBasic |

This disk has electronic catalogs for AMICUS disks 1 to 20 and Fish disks 1 to 80. They are viewed with the DiskCat program, included here.

### AMICUS Disk 22

| | |
|---|---|
| Cycles | Light cycle game, E-D |
| Show_PrintII | Views and prints IFF pictures, including larger than screen |
| PrtDrvGen2.3 | Latest version of a printer driver generator |
| Animations | VideoScape animations of planes and boing ball |
| Garden | Makes fractal gardenscapes |
| BasicSorts | Examples of binary search and insertion sort in AmigaBasic |

### AMICUS Disk 23

| | |
|---|---|
| An AMICUS disk | completely dedicated to music on the Amiga. This disk contains two music players, songs, instruments, and players to bring the thrill of playing "Big Sound" on your Amiga |
| Instruments | a collection of 25 instruments for playing and creating music. The collection ranges from Cannon to Marimba |
| List INSTR | program to list the instruments DMCS will not load as well as list the origins for any instrument. |
| Music | a collection of 14 Classical pieces |
| 1812Overture | The 16 minute classical feature complete with Cannon! |

Three Amiga Music Players:

SMUSPlay
MusicCraft2SMUS
MusicStudio2SMUS

### AMICUS Disk 24

| | |
|---|---|
| Sectorama | A disk sector editor for any AmigaDOS file-structured device, recover files from a trashed hard disk. By David Joiner of MicroIllusions |
| Iconize | Reduces the size of IFF images, companion program, Recolor, remaps the palette colors of one picture to use the palette colors of another. Using these programs and a tool to convert IFF brushes to Workbench icons, make icons look like miniatures of the pictures. |
| CodeDemo | Modula-2 program converts assembler object files to inline CODE statements. Comes with a screen scrolling example |
| AmiBug | Workbench hack makes the same fly walk across the screen at random intervals. Otherwise, completely harmless. |
| BNTools | Three examples of assembly language code from Bryce Nesbitt:<br>1. **SetLace**,prog to switch interlace on&off.<br>2. **Why**, replace AmigaDOS CLI Why<br>3. **LoadIt**, prog to load a file into memory until a reboot. (Only the most esoteric hackers will find LoadIt useful.) |
| Monolace | CLI program resets Preferences to several colors of monochrome & interlace screens. C source is included, works with DisplayPref, a CLI program which displays the current Preferences settings. |
| BoingMachine | A ray-traced animation of a perpetual motion Boing-making machine, includes the latest version of the Movie program, which has the ability to play sounds along with the animation. By Ken Offer |
| Daisy | Example of using the translator and narrator devices to make the Amiga talk. It is written in C. |
| QuickFlix | Script-driven animation and slideshow program flips through IFF images. |
| BMon | System monitor AmigaBasic program ; perform simple manipulations of memory. |
| Moose | Random background program, a small window opens with a moose resembleing Bullwinklesaying witty phrases user definable. |

For PDS orders, Please use form on page 112

| | |
|---|---|
| DGCS | Deluxe Grocery Construction Set, simple Intuition-based prog for assembling and printing a grocery list. |

The Virus Check directory holds several programs relating to the software virus that came to the US from pirates in Europe as detailed in Amazing Computing V2.12. Bill Koester's full explanation of the virus code is included. One program checks for the software virus on a Workbench disk; the second program checks for the virus in memory, which could infect other disks.

### AMICUS Disk 25

| | |
|---|---|
| Nemesis | Graphics demo pans through space towards the mythical dark twin of the sun with wonderful music and space graphics. |

The KickPlay directory holds text that describes several patches to the Kickstart disk. For Amiga 1000 hackers who feel comfortable patching a disk in hexadecimal, KickPlay offers the chance to automatically do an ADDMEM for old expansion memory, as well as the ability to change the picture of the "Insert Workbench" hand. A program is also included for restoring the correct checksum of the Kickstart disk.

| | |
|---|---|
| KeyBird | BASIC prog edits keymaps, adjust the Workbench keymaps or create your own. |
| 8ColorWB | Modifies the Workbench so three bitplanes are used, icons can have eight colors, instead of four, eight-color icons are included. Public domain program "zapicon" or "brush2icon" converts eight-color IFF brushes to icons, to use Deluxe Paint to make icons for this new Workbench. |
| BrushIcon | Converts brushes to icons (bizarr docs). |
| Egraph | Graphing prog reads [x,y] values from a file and displays them on the screen, similar to the same-named Unix program. |
| Keep 1.1 | Message-managing program for telecommunications, lets you save messages from an online transcript to another file, understands the message format of the national networks and several types of bulletin board software. Moves through the transcript and save messages |
| Kill.fastdir | Speed up directory access, it creates a small file in each directory on a disk which contains the information about the files, will also remove all the "fastdir" files from each directory. by CLImate's authors |

The LaceWB program changes between interlace and non-interlace Workbench. Previously, you were forced to reboot after changing Preferences to an interlaced screen. This program flips between the normal and extended screen heights.

| | |
|---|---|
| PW_Utility | A shareware utility for ProWrite users, changes margin settings and font types. |
| Guru | A CLI program, prints out probable causes for Guru mediations; C source included. |
| DiskWipe | Latest from Software Distillery, removes files from directories or disk drives, much faster than "delete." |
| Snow | AmigaBasic makes snowflake designs. |
| Mlist | Mailing list database. |
| Softballstats | Maintain softball statistics/ team records. |
| Dodge | Short Modula-2 program moves the Workbench screen around after a period of time, prevents monitor burn-in. |

### AMICUS Disk 26

Todor Fay's SoundScape module code from his Amazing Computing articles. The source to Echo, Chord, TX, and VU is included. The Lattice and Manx C source code is here, along with the executable modules.

| | |
|---|---|
| ImageMaker | Interesting tool edits Image structures for C, loads & saves C code directly. |
| Claz2 | Update of prog to convert IFF images to PostScript files for printing on laser printers |
| SDBackup | Hard disk backup prog with Lempel-Ziv compression to reduce the necessary number of disks. |
| TCB | Prints information about tasks and processes in the system; assembler source is included. |
| FunBut | Lets a function key act like a rapid series of left mouse button events. |
| DC | A handy program for people who use an Amiga 1020 5 1/4 inch drive as an AmigaDOS floppy. A Workbench program that sends a DiskChange signal to the operating system: instead of typing "diskchange df2:" over and over again, just click on the icon. C source included. |
| System config | File makes screen 80 columns wide of text in the Scribble! word processor. |
| Dick2Ram | 2 programs to move the Scribble! spelling dictionary to and from the RAM disk. |
| Lexical | Analyzes a text file and gives the Gunning-Fog, Flesch, and Kincaid indices which measure readability. |
| HexDump | Modula-2 program to display memory locations in hexadecimal. |
| Tartan | AmigaBasic; design Tartan plaids. |
| DirMaster | Disk catalog program. |
| BMP | plays 8SVX sampled sounds in the background while something else is happening in the Amiga, as your Amiga is booting, for example. |
| ShowPt | CLI program changes your pointer to a given pointer. |

AMICUS 26 also has a collection of mouse pointers, & Workbench program to display them

## *Fred Fish Public Domain Software*

### Fred Fish Disk 1:

| | |
|---|---|
| amigademo | Graphical benchmark for comparing amigas. |
| amigaterm | simple communications program with Xmodem |
| balls | simulation of the "kinetic thingy" with balls on strings |
| colorful | Shows off use of hold-and-modify mode. |
| dhrystone | Dhrystone benchmark program. |
| dotty | Source to the "dotty window" demo on the Workbench disk. |
| freedraw | A small "paint" type program with lines, boxes, etc. |
| gad | John Draper's Gadget tutorial program |
| gfxmem | Graphical memory usage display prog. |
| halfbrite | demonstrates "Extra-Half-Brite" mode, if you have it |
| hello | simple window demo |
| latffp | accessing the Motorola Fast Floating Point library from C |
| palette | Sample prog. to design color palettes. |
| trackdisk | Demonstrates use of the trackdisk driver. |
| requesters | John Draper's requester tutorial and example program. |
| speech | Sample speech demo program. Stripped down "speechtoy". |
| speechtoy | Another speech demo program. |

### Fred Fish Disk 2:

| | |
|---|---|
| alib | Object module librarian. |
| cc | Unix-like frontend for Lattice C compiler. |
| dbug | Macro based C debugging package. Machine independent. |
| make | Subset of Unix make command. |
| make2 | Another make subset command. |
| microemacs | Small version of emacs editor, with macros, no extensions |
| portar | Portable file archiver. |
| xrf | DECUS C cross reference utility. |

### Fred Fish Disk 3:

| | |
|---|---|
| gothic | Gothic font banner printer. |
| roff | A "roff" type text formatter. |
| ff | A very fast text formatter |
| cforth | A highly portable forth implementation. Lots of goodies. |
| xlisp | Xlisp 1.4, not working correctly. |

### Fred Fish Disk 4:

| | |
|---|---|
| banner | Prints horizontal banner |
| bgrep | A Boyer-Moore grep-like utility |
| bison | CNU Unix replacement 'yacc', not working. |
| bm | Another Boyer-Moore grep-like utility |
| grep | DECUS grep |
| kermit | simple portable Kermit with no connect mode. |
| MyCLI | Replacement CLI for the Amiga. V. 1.0 |
| mandel | A Mandelbrot set program, by Robert French and RJ Mical |

### Fred Fish Disk 5

| | |
|---|---|
| cons | Console device demo program with supporting macro routines. |
| freemap | Creates a visual diagram of free memory |
| input.dev | sample input handler, traps key or mouse events |
| joystick | Shows how to set up the gameport device as a joystick. |
| keyboard | demonstrates direct communications with the keyboard. |
| layers | Shows use of the layers library |
| mandelbrot | IFF Mandelbrot program |
| mouse | hooks up mouse to right joystick port |
| one.window | console window demo |
| parallel | Demonstrates access to the parallel port. |
| printer | opening and using the printer, does a screen dump, not working |
| print.support | Printer support routines, not working. |
| proctest | sample process creation code, not working |
| region | demos split drawing regions |
| samplefont | sample font with info on creating your own |
| serial | Demos the serial port |
| singlePlayfield | Creates 320 x 200 playfield |
| speechtoy | latest version of cute speech demo |
| speech.demo | simplified version of speechtoy, with IO requests |
| text.demo | displays available fonts |
| timer | demos timer.device use |
| trackdisk | demos trakcdisk driver |

### Fred Fish Disk 6:

| | |
|---|---|
| compress | like Unix compress, a file squeezer |
| dadc | analog clock impersonator |
| microemacs | upgraded version of microemacs from disk 2 |
| mult | removes multiple occuring lines in files |
| scales | demos using sound and audio functions |
| setparallel | Allows changing parallel port parameters |
| setserial | Allows changing serial port parameters. |
| sortc | quicksort based sort program, in C |
| stripc | Strips comments and extra whitespace from C source |

### Fred Fish Disk 7:

This disk contains the executables of the game Hack V 1.0.1.

### Fred Fish Disk 8:

This disk contains the C source to Hack on disk 7.

### Fred Fish Disk 9:

| | |
|---|---|
| moire | Draws moire patterns in black and white |
| MVP-FORTH | Mountain View Press Forth, version 1.00.03A. A shareware version of FORTH from Fantasia Systems. |
| proff | a more powerful text formatting program |
| setlace | Prog to toggle interlace mode on and off |
| skewb | a rubic's cube type demo |
| sparks | moving snake Graphics demo |

### Fred Fish Disk 10:

| | |
|---|---|
| conquest | An interstellar adventure simulation game |
| dehex | convert a hex file to binary |
| filezap | Patch program for any type of file. |
| fixobj | Strip garbage off Xmodem transferred files. |
| iff | Routines to read and write iff format files. |
| ld | simple directory program |
| ls | Minimal UNIX ls, with Unix-style wildcarding, in C |
| sq,usq | file squeeze and unsqueeze |
| trek73 | Star Trek game |
| yachtc | Dice game. |

### Fred Fish Disk 11:

| | |
|---|---|
| dpslide | slide show program for displaying IFF images with miscellaneous pictures |

### Fred Fish Disk 12:

| | |
|---|---|
| amiga3d | Shows a rotating 3 dimensional solid "Amiga sign". |
| ArgoTerm | a terminal emulator program, written in assembler |
| arrow3d | Shows a rotating 3 dimensional wire frame arrow. |
| ld4 | directory listing program |
| IconExec | |
| SetWindow | two progs for launching progs from Workbench, presently only works under CLI. |
| SetAlternate | Makes an icon show a second image when clicked once |
| StarTerm | terminal emulator, with ASCII Xmodem, dialer, more. |

### Fred Fish Disk 13:

A Bundle of Basic programs, including:

| | | | |
|---|---|---|---|
| Jpad | toybox | ezspeak | mandlebrot |
| xmodem | 3dsolids | addbook | algebra |
| ror | amgseq1 | amiga-copy | band |
| bounce | box | brickout | canvas |
| card1 | circle | colorcircles | Copy |
| cubes1 | cutpaste | datedogstar | |
| dragon | draw | dynamictriangle | |
| Eliza | ezterm | fillibuster | fractal |
| fscape | gomoku | dart | haiku |
| hal9000 | halley | hauntedM | hidden |
| join | loz | mandel | menu |
| minipaint | mouse | Orthello | patch |
| pena | pinwheel | gboxrandom-circles | |
| Readme | rgb | rgbtest | Rord |
| sabotage | salestalk | shades | shapes |
| shuttle | | | |
| sketchpad | spaceart | speakspeach | |
| speecheasy | spell | sphere | |
| spiral | striper | superpad | suprshr |
| talk | terminal | | |
| termtest | tom | topography | triangle |
| wheels | xenos | xmostriper | |

(note: some programs are Abasic, most are Amigabasic, and some programs are presented in both languages)

### Fred Fish Disk 14:

| | |
|---|---|
| amiga3d | update of #12, includes C source to a full hidden surface removal and 3D graphics |
| beep | Source for a function that generates a beep sound |
| dex | extracts text from within C source files |
| dimensions | demonstrates N dimensional graphics |
| filezap | update of disk 10, a file patch utility |
| gfxmem | update of disk 1, graphic memory usage indicator |
| gi | converts IFF brush files to Image struct, in C text. |
| pdterm | simple ANSI VT100 terminal emulator, in 80 x 25 screen |
| shell | simple Unix 'csh' style shell |
| termcap | mostly Unix compatible 'termcap' implementation. |

### Fred Fish Disk 15:

| | |
|---|---|
| Blobs | graphics demo, like Unix 'worms' |
| Clock | simple digital clock program for the title bar |
| Dazzle | An eight-fold symmetry dazzler program. Really pretty! |
| Fish | double buffered sequence cycle animation of a fish |
| Monopoly | A really nice monopoly game written in AbasiC. |
| OkidataDump | Okidata ML92 driver and WorkBench screen dump program. |
| Polydraw | A drawing program written in AbasiC. |
| Polyfractals | A fractal program written in AbasiC. |

### Fred Fish Disk 16:

A complete copy of the latest developer IFF disk

### Fred Fish Disk 17:

The NewTek Digi-View video digitizer HAM demo disk

### Fred Fish Disk 18:

| | |
|---|---|
| AmigaDisplay | dumb terminal program with bell, selectable fonts |
| Ash | Prerelease C Shell-like shell program, history, loops, etc. |
| Browser | wanders a file tree, displays files, all with the mouse |
| MC68010 | docs on upgrading your Amiga to use a MC68010 |
| Multidim | rotate an N dimensional cube with a joystick |
| PigLatin | SAY command that talks in Pig Latin |
| Scrimper | Screen image printer |
| Xlisp1.6 | source, docs, and execut for a Lisp interpret. |

### Fred Fish Disk 19:

| | |
|---|---|
| BlackJack | text-oriented blackjack game |
| JayMinerSlides | Slides by Jay Miner, Amiga graphics chip designer, showing flowchart of the Amiga internals, in 640 x 400. |
| Keymap_Test | test program to test the key mapping routines |
| LockMon | Find unclosed file locks, for programs that don't clean up. |

### Fred Fish Disk 20:

| | |
|---|---|
| AmigaToAtari | converts Amiga object code to Atari form |
| DiskSalv | program to recover files from a trashed AmigaDOS disk. |
| Hash | example of the AmigaDOS disk hashing function |
| Hd | Hex dump utility ala Computer Language magazine, April 86 |
| MandelBrots | Mandelbrot contest winners |
| MultiTasking | Tutorial and examples for Exec level multitasking |
| Pack | strips whitespace from C source |
| PortHandler | sample Port-Handler program that performs. Shows BCPL environment |
| Random | Random number generator in assembly, for C or assembler. |
| SetMouse2 | sets the mouse port to right or left |
| SpeechTerm | terminal Emulator with speech capabilities, XModem |
| TxEd | Demo editor from Microsmith's Charlie Heath |

### Fred Fish Disk 21

This is a copy of Thomas Wilcox's Mandelbrot Set Explorer disk. Very good!

### Fred Fish Disk 22

This disk contains two new "strains" of microemacs.

| | |
|---|---|
| Lemacs | version 3.6 by Daniel Lawrence. For Unix V7, BSD 4.2, Amiga, MS-DOS, VMS. Uses Amiga function keys, status line, execute, startup files, more. |
| Pemacs | By Andy Poggio. New features include <ALT> keys as Meta keys, mouse support, higher priority, backup files, word wrap, function keys. |

### Fred Fish Disk 23

Disk of source for MicroEmacs, several versions for most popular operating systems on micros and mainframes. For people who want to port MicroEmacs to their favorite machine.

### Fred Fish Disk 24:

| | |
|---|---|
| Conques | interstaller adventure simulation game |
| Csh | update to shell on Disk 14, with built in commands,named variables substitution. |

Modula-2 A pre-release version of the single pass Modula-2 compiler originally developed for Macintosh at ETHZ. This code was transmitted to the AMIGA and is executed on the AMIGA using a special loader. Binary only.

### Fred Fish Disk 25

| | |
|---|---|
| Graphic Hack | A graphic version of the game on disks 7 and 8 This is the graphics-oriented |
| Hack | game by John Toebes. Only the executable is present. |

### Fred Fish Disk 26

UnHunk Processes the Amiga "hunk" loadfiles. Collect code, data, and bss hunks together, allows individual specificatio of code, data, and bss origins, and generates binary file with format reminiscent of Unix "a.out" format. The output file can be easily processed by a separate program to produce Motorola "S-records" suitable for downloading to PROM programmer. By Eric Black.

| | |
|---|---|
| C-kermit | Port of the Kermit file transfer program and server. |
| Ps | Display and set process priorities |
| Archx | Yet another program for bundling up text files and mailing or posting them as a single file unit. |

### Fred Fish Disk 27

| | |
|---|---|
| ABdemos | Amiga Basic demos; Caroly Scheppner. |
| NewConvertFD | creates .bmaps from fd files. |
| BitPlanes | finds addresses of and writes to bitplanes of the screen's bitmap. |
| AboutBmaps | A tutorial on creation and use of bmaps. |
| LoadILBM | loads and displays IFF ILBM pics. |
| LoadACBM | loads and displays ACBM pics. |
| ScreenPrint | creates a demo screen and dumps it to a graphic printer. |
| Disassem | Simple 68000 disassembler. Reads standard Amiga object files and disassembles the code sections. Data sections are dumped in hex. The actual disassember routines are set up to be callable from a user prog so instructions in memory can be disassembled dynamically. By Bill Rogers. |
| DvorakKeymap | Example of a keymap structure for the Dvorak keyboard layout. Untested but included because assembly examples are few and far between. By Robert Burns |
| Hypocycloids | Spirograph, from Feb. 84 Byte. |
| LinesDemo | Example of proportional gadgets to scroll a SuperBitMap. |
| MemExpansion | Schematics and directions for building your own homebrew 1 Mb memory expansion, by Michael Fellinger. |
| SafeMalloc | Program to debug 'malloc()' calls |
| ScienceDemos | Convert Julian to solar and sidereal time, stellar positions and radial velocity epoch calculations and Galilean satellite plotter. By David Eagle. |

### Fred Fish Disk 28

ABasic games by David Addison: Backgammon , Cribbage, Milestone, and Othello

| | |
|---|---|
| Cpp | DECUS 'cpp' C preprocessor, & a modified 'cc' that knows about the 'cpp', for Manx C. |
| Shar | Unix-compatible shell archiver, for packing files for travel. |
| SuperBitMap | Example of using a ScrollLayer, syncing SuperBitMaps for printing, and creating dummy RastPorts. |

### Fred Fish Disk 29

| | |
|---|---|
| AegisDraw Demo | Demo program without save and no docs. |
| Animator Demo | Player for the Aegis Animator files |

| | |
|---|---|
| Cc | Unix-like front-end for Manx C |
| Enough | Tests for existence of system resources, files, and devices |
| Rubik | Animated Rubik's cube program |
| StringLib | |
| Vt100 | VT-100 terminal emulator with Kermit and Xmodem protocols |

**Fred Fish Disk 30**

Several shareware programs. The authors request a donation if you find their program useful, so they can write more software.

| | |
|---|---|
| BBS | an Amiga Basic BBS by Ewan Grantham |
| FineArt | Amiga art |
| FontEditor | edit fonts, by Tim Robinson |
| MenuEditor | Create menus, save them as C source, by David Pehrson |
| StarTerm3.0 | Very nice telecom by J. Nangano |

(Fred Fish Disk#30 is free if requested when ordered with at least three other disks from the collection.)

**Fred Fish Disk 31**

| | |
|---|---|
| Life | Life game, uses blitter to do 19.8 generations a second. |
| Mandelbrot | Version 3.0 of Robert French's program. |
| MxExample | Mutual exclusion gadget example. |
| RamSpeed | Measure relative RAM speed, chip and fast. |
| Set | Replacement for the Manx "set" command for environment variables,with improvements. |
| Tree | Draws a recursive tree, green leafy type, not files. |
| TxEd | Crippled demo version of Microsmith's text editor, TxEd. |
| VDraw | Full-featured drawing program by Stephen Vermeulen. |
| Xicon | Invokes CLI scripts from icon |
| Ticon | Displays text files from an icon. |

**Fred Fish Disk 32**

| | |
|---|---|
| Address | Extended address book, AmigaBASIC |
| Calendar | Calendar/diary program, AmigaBASIC |
| DosPlus1 | First volume of CLI oriented developer tools |
| DosPlus2 | 2nd volume of CLI oriented developer tools. |

Executables only:

| | |
|---|---|
| MacView | Views MacPaint pics in Amiga low or high res, no sample pictures, by Scott Evernden. |
| Puzzle | Simulation of puzzle with moving squares. |
| ShowHAM | View HAM pictures from CLI. |
| Solitaire | ABasiC games of Canfield and Klondike, from David Addison. |
| Spin3 | Graphics demo of spinning cubes, double-buffered example. |
| Sword | Sword of Fallen Angel text adventure game written in Amiga Basic. |
| Trails | Leaves a trail behind mouse, in Modula-2 |

**Fred Fish Disk 33**

| | |
|---|---|
| 3dstars | 3d version of the "stars" program below. |
| Bigmap | Low-level graphics example scrolls bitmap with ScrollVPort. |
| Dbuf.gels | Double-buffered animation example for BOBs and VSprites. |
| DiskMapper | Displays sector allocation of floppy disks. |
| MemView | View memory in real time, move with joystick. |
| Oing | Bouncing balls demo |
| Sproing | Oing, with sound effects. |
| ScreenDump | Dumps highest screen or window to the printer. |
| Sdb | Simple database program from a DECUS tape. |
| Stars | Star field demo, like Star Trek. |
| TermPlus | Terminal program with capture, library, function keys, Xmodem, CIS-B protocols. |
| Vt100 | Version 2.0 of Dave Wecker's VT-100 emulator, with scripts & function |

**Fred Fish Disk 34**

| | |
|---|---|
| Alint | Support files for Gimpel's 'lint' syntax checker |
| Blink | PD 'alink' compaible linker,faster,better. |
| Browser | Updated to FF 18 'browser', in Manx, with scroll bars, bug fixes. |
| Btree | b-tree data structure examples |
| Btree2 | Another version of 'btree' |
| Calendar | Appointment calendar with alarm. |
| Less | File viewer, searching, position by percent, line number. |
| NewFonts | Set of 28 new Amiga fonts from Bill Fischer |
| Pr | Background print utility, style options, wildcards. |
| Requester | Deluxe Paint-type file requester, with sample. |

**Fred Fish Disk 35**

| | |
|---|---|
| ASendPacket | C example of making asynchronous I/O calls to a DOS handler, written by C-A |
| ConsoleWindow | C example of getting the Intuition pointer a CON: or RAW: window, for 1.2, by C-A. |
| DirUtil | Walk the directory tree, do CLI operations from menus |
| DirUtil2 | Another variant of Dirutil. |
| FileRequester | Lattice C file requester module, with demo driver, from Charlie Heath. |
| MacView | Views MacPaint pictures in Amiga low or high res, with sample pictures, by Scott Evernden. |
| Plop | Simple IFF reader program |
| PopCLI | Sidekick-style program invokes a new CLI, with automatic screen blanking. |
| QuickCopy | Devenport disk copiers duplicate copy-protected disks. |
| ScrollPf | Dual playfield example, from C-A, shows 400 x 300 x 2 bit plane playfield on a 320 x 200 x 2 plane deep playfield. |
| SendPacket | General purpose subroutine to send AmigaDos packets. |
| SpriteMaker | Sprite editor, can save work as C data structure. Shareware by Ray Larson. |
| Tracker | Converts any disk into files, for electronic transmission. Preserves entire file structure. Shareware by Brad Wilson. |
| TriClops 3-D | space invasion game, formerly commercial, now public domain. From Geodesic Publications. |
| Tsize | Print total size of all files in subdirectories. |
| UnIfdef | C preprocessor to remove given #ifdef'd sections of a file, leaving the rest alone. By Dave Yost |
| Vttest | VT-100 emulation test program. Requires a Unix system. |

**Fred Fish Disk 36**

| | |
|---|---|
| Acp | Unix-like 'cp' copy program |
| Clock | Updated version of clock on disk 15. |
| Csh | Manx 'csh'-like CLI, history, variables, etc. |
| DietAid | Diet planning aid organizes recipes, calories |
| Echo | Improved 'echo' command with color, cursor addressing |
| FixHunk | Fixs programs to let them run in external memory. |
| Fm | Maps the sectors a file uses on the disk. |
| KickBench | Docs, program to make a single disk that works like a Kickstart and Workbench. |
| Lex | Computes Fog, Flesch, and Kincaid readability of text files. |
| TunnelVision | David Addison ABasic 3D maze perspective game. |
| Vc | Visicalc-like spreadsheet calculator program. |
| Vt100 | Version 2.2 of Dave Wecker's telecom program |
| YaBoing | Oing! style game program shows sprite collision detects |

**Fred Fish Disk 37**

This disk is a port of Timothy Budd's Little Smalltalk system, done by Bill Kinnersley at Washington State University.

**Fred Fish Disk 38**

| | |
|---|---|
| CSquared | Sep 86 Sci American, Circle Squared algorithm |
| FixObj | Strips garbage off Xmodem transfered object files |
| Handler | AmigaDOS handler (device) example from C-A |
| Hp-10c | Mimics a HP-10C calculator, written in Modula-2 |
| IFFEncode | Saves the screen as an IFF file |
| IffDump | Dumps info about an IFF file |
| Jsh | BDS C-like CLI shell |
| NewStat | STATUS-like program, shows priority, processes |
| Reversi | Game of Reversi, version 6.1 |
| UUdecode | Translate binary files to text, Unix-like programs |
| Vdraw | Drawing program, version 1.14 |
| VoiceFiler | DX MIDI synthesizer voice filer program |
| Window | Example of creating a DOS window on a custom screen |

**Fred Fish Disk 39**

| | |
|---|---|
| AnsiEcho | 'echo', 'touch', 'list', 'cls' written in assembler. |
| Display | Displays HAM images from a ray-tracing program, with example pictures. |
| Driver | Example device driver source, acts like RAM: disk |
| Xlisp | XLisp 1.7, executable only |

**Fred Fish Disk 40**

| | |
|---|---|
| Ahost | Terminal emulator with Xmodem, Kermit and CIS B protocols, function keys, scripts, RLE graphics and conference mode. |
| AmigaMonitor | Dynamically displays the machine state, such as open files, active tasks, resources, device states, interrupts, libraries, ports, etc. |
| Arc | Popular file compression system, the standard for transitting files |
| AreaCode | Program that decodes area codes into state and locality. |
| Blink | 'alink' replacement linker, version 6.5 |
| Cosmo | An 'asteriods' clone. |
| Dg210 | Data General D-210 Terminal emulator |
| DirUtil | Windowed DOS interface program, V 1.4 |
| DOSHelper | Windowed AmigaDOS CLI help program |
| PagePrint | Prints text files with headers, page breaks, line numbers |
| PopCLI | Starts a new CLI with a single keystroke, from any program, With a screen-saver feature. Version 2 with |
| SpriteEd | Sprite Editor edits two sprites at a time |
| X-Spell | Spelling checker allows edits to files |

**Fred Fish Disk 41**

| | |
|---|---|
| AmigaVenture | Create your own text adventure programs in AmigaBasic. |
| Csh | Version 2.03 of Dillon's C sh-like shell. Executable only |
| Doug | Macro based C debugging package, FF #2 |
| DualPlayField | example from CBM, update to Intuition manual |
| GetFile | Heath's file requester, with source |
| LatXref | Cross reference of Lattice 3.10 header files |
| Lines | Line drawing demo program |
| SetFont | Changes font used in a CLI window |
| Vt100 | Version 2.3 of the VT-100 terminal program. |

**Fred Fish Disk 42**

This disk contains an Amiga version of MicroGNUEmacs.

**Fred Fish Disk 43**

| | |
|---|---|
| BasicBoing | AmigaBasic program demos page flipping of a 3D cube |
| Bbm | Demo copy of B.E.S.T. Business Management System. |
| BbsList | A list of Amiga Bulletin Board Systems |
| Cc | C compiler frontends for Manx and Lattice C |
| Copper | A hardware copper list disassembler |
| InstIFF | Converts Instruments demo sounds to IFF sampled sounds |
| PopColours | Adjust RGB colors of any screen |
| SpriteClock | Simple clock is displayed on a sprite above all screens |
| ST Emulator | Non-serious Atari ST emulator |
| WBrun | Lets Workbench programs be run from the CLI |
| Wild | Two Unix shell style wild card matching routines |

**Fred Fish Disk 44**

| | |
|---|---|
| Icons | Miscellaneous icons |
| NewIFF | New IFF material from CBM for sampled voice and music files |
| RayTracePics | The famous ray-tracing pictures, from FF#39, now converted to IFF HAM |
| format | for "much" faster viewing. |
| ViewILBM | Displays normal and HAM ILBM files |

**Fred Fish Disk 45**

| | |
|---|---|
| Clue | Clue board game |
| Make | Another 'make', with more features |
| Pictures | Miscellaneous pictures |
| Update | Updates an older disk with newer files from another disk |
| WhereIs | Searches a disk for files of given name |

**Fred Fish Disk 46**

| | |
|---|---|
| Asm | Shareware 68010 macro assembler, ROM Kernal Manual compatible |
| CheckModem | 'execute' file program detects presence of modem |
| Egad | Gadget editor from the Programmers Network |
| Jive | Transforms a file from English to Jive. |
| My.lib | A binary only copy of Matt's alternate runtime library. Author: Matt Dillon |
| ProffMacros | Subset Berkeley 'ms' and 'mm' macros for 'proff' |
| ValSpeak | Transforms a file from English to Valley Speak. |

**Fred Fish Disk 47**

| | |
|---|---|
| 3D-Arm | Simulation of a robotic arm, very good graphics, teaching tool, including C source. |
| Juggler | Eric Graham's stunning HAM animation of a robot juggler |
| VT-100 | Version 2.4 of Dave Wecker's terminal emulator, with Xmodem and Kermit file transfer protocols |

**Fred Fish Disk 48**

| | |
|---|---|
| Bru | Alpha version of a hard disk file archiver |
| Comm | Version 1.30 of a terminal emulator with phone directories |
| Csh | Version 2.04 of Matt Dillion's Unix 'csh'-like CLI replacement, including Lattice and Manx C source |
| Diskperf | Disk benchmark program for Unix and Amiga |
| Du | Computes disk storage of a file or directory |
| MemWatch | Program to watch for programs that trash low memory. It attempts to repair the damage, and puts up a requester to inform you of the damage. From the Software Distillery. |
| Profiler | A realtime execution profiler for Manx C programs. Includes C source. |

**Fred Fish Disk 49**

| | |
|---|---|
| Cycloids | Update of electronic spirograph from disk 27 |
| DirUtil | Enhanced version of DirUtil from disk 35 |
| MultiDef | Scans a set of object modules and libraries searching for multiply defined symbols |
| MyUpdate | Disk update utility with options for stripping comments from C header files, and interactive verification of the updating process |
| Plot | Computes and displays 3 dimensional functions in hires |
| Polygon | Moire type pattern generator with color cycling |
| QMouse | Queries whether a mouse button is pressed. This can give a return code that can customize a startup-sequence based on whether a mouse button was pressed. |
| Touch | Example of setting the datestamp on a file, using a technique from Commodore-Amiga |
| Trees | More extensive version of the trees program on Disk 31 |

**Fred Fish Disk 50**

| | |
|---|---|
| Asm | Version 1.1 of a shareware 68000 macro assembler, compatible with the Metacomco assembler. This includes an example startup module and more Motorola mneumonics. |
| BreakOut | A brick breakout game, uses 3-D glasses |
| DiskZap | Version 1.1 of a program to edit disks and binary files |
| FirstSilicon | A smart CLI replacement with full editing and recall of previous commands |
| Missle | A Missile Command-type game, with sound, in assembler |
| PerfectSound | Sound editor for a low-cost sound digitizer |
| Sizzlers | Graphics demos |
| UnixArc | Ver of 'arc' for Unix System V machines,in C |
| Wombat | Version 3.01 of Dave Warker's terminal emulator |

**Fred Fish Disk 51**

| | |
|---|---|
| Bison | GNU for Unix 'yacc', working update to FF4 |
| Compress | Update to the file compression program on Disk 6 |
| Cos | "Wheel of Fortune"-type game in AmigaBasic |
| DifSsed | Unix-like 'dif' and 'ssed' for finding the differences between two files, and then recreating the other, given one file, and the list of differences. |
| Sq, Usq | Portable versions of the CP/M squeeze and unsqueeze |

**Fred Fish Disk 52**

| | |
|---|---|
| Assign | Replacement for AmigaDOS 'assign' command in C |
| Fractal | Makes random fractal terrains |
| Poly, HAMPoly | Workbench-type demos for making polygons in lores and HAM |
| MxGads | Example of mutual exclusion gadgets with GadgetText |
| Tek4010 | Tektronix 4010 terminal emulator |
| VDraw | Versions 1.16 and 1.19 of a Deluxe Paint-like drawing program |

**Fred Fish Disk 53**

| | |
|---|---|
| Animations | Demo animations with player program for Aegis Animator |
| ARCre | Creates rename scripts for files with long names, so they can be easily 'arc'ed and un'arc'ed. |
| ARP | Preliminary AmigaDOS replacements for 'break', 'cd', 'chmod', 'echo', 'filenote' and ' makedir' |
| Compiler | Not fully ported to the Amiga, this is a 68000 C compiler. It wil produce simple assembly language output, but needs a lot of work. |
| Spreadsheet | Update with source of the 'vc' spreadsheet on disk 36 |
| TarSplit | Port of program to split Unix 'tar'archives |
| UUencode | Utilties to encode and decode binary files for ASCII transmission, expanding them by 35% |

**Fred Fish Disk 54**

| | |
|---|---|
| Hanoi | Solves Towers of Hanoi Problem in it's own Workbench window. by Ali Ozer |
| ISpell | Port of a Unix screen oriented, interactive spelling checker. (Expansion RAM required) by Pace Willisson |
| Ing | A Screen of lots of bouncing little windows by Leo 'Bols Ewhac' Schwab |
| Lav | Displays number of tasks in run queue, averaged over last 1, 5, and 15 minute periods. by Willaim Rucklidge |
| MIDITools | Programs to play/record through the MIDI I/F. by Fred Cassirer |
| MoreRows | Program to make the Work Bench Screen larger than normal. by Neil Katin and Jim Mackraz |
| Tilt | Program to make your Amiga look like it didn't pass vibration testing. by Leo ' Bols Ewhac' Schwab |

**Fred Fish Disk 55**

| | |
|---|---|
| Csh | V2.05 of Matt Dillon's csh like shell (Modified for Manx C). by Matt Dillon, Modified by Steve Drew |
| NewStartups | New C Startup modules: |
| AStartup.asm | with 1.2 fixes and better quote handling. |
| TWStartup.asm | opens a stdio window, using user specs. by Commodore, posted to BIX by Carolyn Schepper |
| Palette | Change another program's screen colors. by Carolyn Schepper |
| PipeDevice | Allows the standard output of one process to be fed to the standard input of another. by Matt Dillon |
| ScreenSave | Save a normal or HAM mode screen as an IFF file. by Carolyn Schepper |
| ShanghaiDemo | Demo of the Activision game Shanghai. |
| SoundExample | A double buffered sound example for Manx C. by Jim Goodnow |
| Vsprites | A working vsprite example. by Eric Cotton |
| Vt100 | V2.6 of Dave's Vt100 terminal emulator with kermit and xmodem. by Dave Wecker |

**Fred Fish Disk 56**

| | |
|---|---|
| ClipBoard | Clipboard device interface routines, to provide a standard interface. by Andy Finkle |
| ConPackets | Demos the use of DOS Packets, ConUnit, etc. by Carolyn Schepper |
| GetDisks | Program to find all available disk device names and return them as an exec list. by Philip Lindsay |
| GetVolume | Program to get volume name of the volume that a given file resides on. by Chuck McManis |
| Icon2C | Reads an icon file and writes out a fragment of C code with the icon data structures. by Carolyn Schepper |
| MergeMem | Program to merge the MemList entries of sequentially configured RAM boards. by Carolyn Schepper |
| mCAD | An object oriented drawing program, V1.1 by Tim Mooney |

**Fred Fish Disk 57**

Replaced by FF97 Due to Copyright problems

**Fred Fish Disk 58**

| | |
|---|---|
| ASDG-rrd | Extremely useful shareware recoverable ram disk. by Perry Kivolowitz |
| BigView | Displays any IFF picture, independent of the physical display size, using hardware scroll. by John Hodgson |
| EGraph | Reads pairs of x and y value from a list of files and draws a formatted graph. by Laurenece Turner |
| HyperBase | Shareware data management system. V1.5 |
| MemClear | Walks through the free memory lists, zeroing free memory along the way. by John Hodgson |
| NewZAP | A third-generation multi-purpose file sector editing utility. V3.0 by John Hodgeson |
| RainBow | A Maurauder-Style rainbow generator. by John Hodgson |
| SMUSPlayers | Two SMUS plays, to play SMUS IFF music formatted files. by John Hodgson |
| View | A tiny ILBM viewer by John Hodgson |
| WBdump | JX-80 optimized workbench printer that does not use DumpRPort. by John Hodgson |

For PDS orders, Please use form on page 112

### Fed Fish Disk 59

| | |
|---|---|
| Browser | Update to browser program on disks 18 and 34. S-E |
| Browser2 | Another different browser program. E |
| Clock | Clock program with fonts, colors. E |
| Dme | Dillon text editor V1.22 for programmers,ED |
| DropCloth | Puts pattern on Workbench backdrop,ED |
| DropShadow | Puts shadows on Workbench windows,E-D |
| FixWB | Similar to DropCloth, but doesn't work yet. S-D |
| mCAD | Object-oriented drawing program, version 1.2.2. Much improved over disk 56. |
| Robotroff | Demo of animated pointers on Workbench. S-E-D |
| Supermort | General compounding/amortization loan calculator. E-D |

### Fred Fish Disk 60

Various shareware and freeware programs

| | |
|---|---|
| Blitz | Memory resident file viewer. Very fast. E-D |
| BlitzFonts | Makes text output faster. E-D |
| HandShake | Terminal emulator with VT52/VT100/ VT102support. E-D |
| Med | Mouse-driven text editor version 2.1. E-D |
| PrtDrvGen | Generates printer drivers, version 1.1.S available from author. E-D |
| Show | Slideshow-like IFF viewer, V2.1. E-D |
| Uedit | Customizable text editor V2.0. E-D |
| Ueturbo | Example Uedit setup macros. S-E-D |

### Fred Fish Disk 61

| | |
|---|---|
| ATPatch | Patches Transformer to work under AmigaDOS 1.2. S-E-D |
| FillDisk | Writes zeroes to free blocks on a disk for security. S-E-D |
| LPatch | Patch for programs that abort when loading under AmigaDOS 1.2. S-E-D |
| MicroEmacs | Conroy MicroEmacs V3.8b, newer than disk 22. S-E-D |
| PearlFont | Like Topaz, but rounded edges. |
| Terrain | Generates fractal scenery. S-E-D |
| VSprites | Makes 28 Vsprites, from S-E-D book. |

### Fred Fish Disk 62

This is a port of the Unix game 'Hack', by the Software Distillery, version 1.0.3D.

### Fred Fish Disk 63

This is a port of the Unix game 'Larn', by the Software Distillery, version 12.0B.

### Fred Fish Disk 64

This is an offical IFF specification disk from Commodore, an update to disk 16.

### Fred Fish Disk 65

| | |
|---|---|
| Bawk | Unix text processor, like 'awk'. Doesn't work, but source is included. S-E-D. |
| MWB | Example of rerouting Workbench window open calls to another custom screen. Version 1.01, S-E-D |
| CloseWB | Example for closing a custom Workbench screen. S-E-D |
| Cookie | Generates one-line fortune-cookie aphorisms. S-E-D |
| JTime | Build-your-own mouse port clock. |
| MenuBuilder | Creates C source files for menus, based on text descriptions. S-E-D. |
| NewPackets | CBM tutorial on new packets and structures in AmigaDos 1.2. |
| PascalToC | Pascal to C translator, not so great. S-E-D |
| Prep | 'ratfor'-like FORTRAN preprocessor. S-E-D |
| RunBack | Starts programs from CLI, allowing CLI window to close. E-D |
| SunMouse | This program automatically clicks in windows when the mouse is moved over them. Version 1.0, E-D |

### Fred Fish Disk 66

| | |
|---|---|
| AmScsi | Preliminary plans for a SCSI disk controller board. |
| Asm68k | Macro assembler, version 1.0.1. E-D |
| Assigned | Example for avoiding DOS insert-disk requester, by scanning the list of 'assign'ed names. S-E-D |
| Dk | Pretends to eat away at CLI window. S-E-D |
| Flip | Flips whole screen as a joke. S-E-D |
| Foogol | Foogol cross-compiler generates VAX assembly code. S-E-D |
| Free | Prints amount of free space on all drives. S-E-D |
| MallocTest | malloc/free memory test program. S-E-D |
| Melt | Pretends to melt the screen. S-E-D |
| Nart | Graphic flying string demo. S-E-D |
| Purty | Easy way to set printer attributes from Workbench. E-D |
| RayTracer | Simple ray tracing program. E-D |
| SendPackets | Updated CBM examples of packet routines on disk 35. S-E-D |
| SnapShot | Memory resident screen dump. E-D |
| TagBBS | Shareware BBS system, version 1.02. |

### Fred Fish Disk 67

| | |
|---|---|
| AmCat | Shareware disk catalogi E-D program. |
| AmigaSpell | Shareware Intuition spelling checker, V2.0. E-D |
| Bouncer | 3-D bouncing ball written in MultiForth, SED |
| Comm | Terminal program version 1.33, E |
| Dux5 | Another version of DirUtil. S-E-D |
| HexCalc | Hex, octal, & decimal calculator. E-D |
| Icons | Various big and alternate image icons. |
| Mandala | Mandala graphics and sound. E |
| PersMait | Demo shareware personal file manager. |
| RSLClock | Menu bar clock version 1.3. E-D |
| RTCubes | Graphics demo of 3D cubes. E-D |
| Wheel | "Wheel of Fortune"-type game,AmigaBASIC |

### Fred Fish Disk 68

This is version MG 1b of the MicroGNUEmacs. Source and executable areincluded, as well as source for other computers besides the Amiga.

### Fred Fish Disk 69

| | |
|---|---|
| Asm68k | Macro assembler, v1.0.3, E-D |
| BlitLab | Blitter exploring program, in C, S-E-D |
| Conman | Replacement console device handler adds editing and history to any application that uses CON:, v0.9, E-D |
| Console | Replacement console routines, in C, S-E-D |
| Dk | Decays the screen bit by bit, update to disk 66, in Modula-2, S-E-D |
| Frags | Displays memory fragmentation listing the size of free memory blocks, in C, S-E-D |
| IconType | Change the type of an icon, in C, S-E-D |
| Make | 'make' in Manx C, S-E-D |
| MonProc | Monitors processes for packet activity, in C,S-ED |
| MouseClock | Mouse pointer into a digital clock,in C,SED |
| Sb | Browses system structures, from Transactor magazine, v1.0, in C, S-E-D |
| Spew | Generates 'National Enquirer'-type headlines from rules file. In C,S-E-D |
| Spool | Three programs to demonstrate multitasking and spooling in a printer spooler. In C, v1.2, S-E-D |
| Wc | Counts words ala Unix 'wc', but faster, in C,S-ED |

### Fred Fish Disk 70

This is a disk of shareware programs.

| | |
|---|---|
| AmigaMonitor | Explores state of the system, v1.13 |
| Arc | Standard file compressor andl librarian, v0.23, a port of MS-DOS v5.0. E-D |
| BlackBook | Phone book program. |
| DoTil | Intuition-driven file manipulator program,v2.0. |
| GravityWars | Game of planets, ships and black holes,v1.03. |
| Jobs | Alternate user interface to CLI and WB, v2.1. |
| Lens | Magnifies area around mouse, shows it in a window, v1.0. |
| Life-3d | 3D version of the classic cellular-automaton game, v1.2. |
| Logo | Logo language interpreter |
| SetKey | Demo keymap editor, v1.0 |
| Vpg | Makes displays for aligning video monitors, v1.0. |

### Fred Fish Disk 71

| | |
|---|---|
| AirFoil | Makes airfoils using the Joukowski transformation, in C, S-E-D |
| Amiga Basic | Miscellaneous programs including 3D plot program, a kaleidoscope, C-A logo drawing program file comparison utility string search program, S-E-D |
| Blocks | A variation of "lines", but with variable color blocks. E-D |
| Comm | Great terminal program, v1.34, E-D |
| DiskX | Utility for exploring file system.E-D |
| Fpic | Simple image processing program that operates on IFF pictures, with several filters, merging images, E-D |
| IconMk | Makes icons for files , v1.2a, E-D |
| Icons | New icons |
| NewFonts | Two new fonts; 'shalt18', an electronic ciruit element font, and 'ibm5', a PC-like font. |
| PetCLI | An AmigaBASIC CLI shell program. |
| PWDemo | Demo of the commercial product PowerWindows,v1.2. It aids creation of custom windows, menus, and gadgets, giving C or assembly source. E-D |
| Rot | Creates and animates 3-D objects, v0.5, E,D |
| TimeSet | Sets time from Workbench, E-D |

### Fred Fish Disk 72

This is a disk of IFF pictures.

### Fred Fish Disk 73

| | |
|---|---|
| Add | Customizes existing program menus with Amiga-key shortcuts. Also includes 'until', which waits until a given window is created. Shareware, in C, S-E-D. |
| AutoIconOpen | Fools WB into thinking mouse has double-clicked icons. In C, S-E-D |
| Dio | Generic Exec device interface code for opening libraries, getting multiple I/O channels, asynchronous operations, etc. In C, S-E-D. |
| Dissolve | Slowly displays IFF files, ala Nov 86 Dr. Dobb's program. In C, S-E-D |
| DTerm | Flexible, reprogrammable terminal program v1.10, E-D |
| Expose | Re-arranges windows so that at least one pixel of menu bar gadgets are exposed. In C, S-E-D. |
| Lit | Scans a text file, converts to C-style printable strings.C,v2.0, S-E-D |
| Lmv | "Long Movie", program views series of IFF picts in quick succession, upto 19 fps. Shareware, E-D |
| MouseOff | Mouse pointer disappears after ten seconds of non-use. In C, S-E-D |
| ParOut | Examples of controlling parallel port with resources instead of the PAR: device. In C,S-E-D |
| PenPalFont | Child-like font. |
| RunBackGround | Similar to RunBack on disk 66, runs program from the CLI allowing the CLI window to close. In C,S-E-D |
| SnapShot | Screendump utility,update FF 66.E-D |
| TypeAndTell | Example installs a device handler before Intuition, and speaks each key as it is pressed. In C and assembler, S-E-D |
| Xplor | Prints info about system lists, in assembler,S-E-D |

### Fred Fish Disk 74

| | |
|---|---|
| Cled | Edits and recalls CLI commands, v1.3, E-D |
| Control | Intercepts graphic printer dump calls and accesses color map, width,and screen resolution. C,S-E-D |
| Dme | Simple WYSIWYG text editor for programmers,v1.25. Update of FF 59.E-D |
| DropShadow | WB dropshadows, v2.0.UpdateFF59. E-D |
| Funds | AmigaBASIC prog tracks mutual or stocks-D |
| Less | Text viewing program, like Unix "more", v1.1, update to disk 34. S-E-D |
| Makemake | Scans C source files and constructs a vanilla 'makefile' in the current directory. S-E-D |
| mCAD | Object-oriented drawing prog. v1.2.4, update to FF 59.Shareware, E-D |
| Random | Simple random number generator in C. S-E-D |
| TDebug | Monitors devices by intercepting Exec SendIO() and DoIO() vectors, in C, v1.0, S-E-D |
| Units | Converts measurements in different units, includes "chart" option, in C, S-E-D |
| XCopy | Replacement for AmigaDOS 'copy', doesn't change the date, uses Unix wildcards. E-D |

### Fred Fish Disk 75

| | |
|---|---|
| Bezier | Play with Bezier curves points and granularity, S-E-D |
| BSplines | Play with b-splines, as above, S-E-D |
| Comm | C source for Comm terminal program v1.34. S-E-D |
| Copy | Replacement 'copy' command v1.0, preserves date, in C, S-E-D |
| Diff | Simple 'diff' in C, S-E-D |
| DuM2 | Another DirUtil in Modula-2, v1.5, S-E-D |
| Eless | Fast 'dir' program in C, S-E-D |
| Fd | Faster 'eless' in C, S-E-D |
| HardCopy | Sends a transcript of a CLI session to a file, in C, S-E-D |
| MouseOff | Update FF73, turns off mouse pointer, S-E-D |
| SetFont | Changes the font in a Workbench screen, v2.0, S-E-D |
| SpeedDir | Another fast 'dir', in assembler, S-E-D |

### Fred Fish Disk 76 & 77

These are disks 1 and 2 of Chris Gray's Draco distribution for the Amiga. Draco is a compiled, structured language reminiscent of both C and Pascal. A full interface to AmigaDOS and Intuition is supplied. Be sure to get both disk 76 and 77.

### Fred Fish Disk 78

| | |
|---|---|
| Cycles | Cycle game like "Tron", v1.0, E-D |
| EOMS | Experts Only Mercenary Simulator game, E-D |
| MandelVroom | Mandelbrot generator with enhanced palette controls, fixed/floating point, presets, v1.50, in Manx C, S-E-D |

### Fred Fish Disk 79

| | |
|---|---|
| AsmTools | CLI tools in assembler: echo, loadit, mounted, setlace, why; S-E-D |
| AssignDev | Give devices multiple names, in C, S-E-D |
| AuxHandler | Example of a dos handler that allows use of a CLI via the serial port. Includes source. Author: Steve Drew |
| Cmd | Redirects printer output to a file, in C, S-E-D |
| Info | AmigaDOS 'info' replacement, in C and assembler, S-E-D |
| Kill | Removes a task and its resources, in C,S-E-D |
| M2Error | Displays errors from TDI Modula-2 compiles, S-E-D |
| MonProc | Update to process packet program from disk 69, in C, S-E-D |
| Mounted | Program for testing if a drive is present, in a script In C, S-E-D |
| Nro | Another 'roff'-style text formatter, in C, S-E-D |
| ParTask | Finds parent task, in C, S-E-D |
| QueryAny | For scripts, asks a question, accepts Y/N, gives return code. In assembler, S-E-D |
| ScnSizer | Resets pref settings for screen size, in C,SED |
| SharedLib | Example, shared lib, in C & assembler, S-E-D |
| Task | Simple CreateTask() example in C, S-E-D |
| Uw | Unix Windows client v1.0, in C, S-E-D |
| Who | Lists tasks on ready and wait queues, in C, S-E-D |

### Fred Fish Disk 80 (see Fred Fish 90)

Fred Fish 80 has been withdrawn due to copyright problems.

### Fred Fish Disk 81

| | |
|---|---|
| Asm68k | V1.1.0 of a macro assembler |
| AutoFacc | Shrinks the FACC window and moves it to the back |
| Brushes | 53 custom IFF brushes of electronic symbols |
| CheckIFF | Checks structure of an IFF file CledV1.4 update FF74 of a simple CLI |
| Conman | Replaces console handler to add editing and history to many programs |
| Fonts | Miscellaneous fonts |
| Icon | V6.0 of the Icon programming language |
| KeyLock | Freezes the keyboard and mouse until pass word entered. |
| ScatDisplay | hack created from "lng" |
| Smush | Smushes an IFF file. |
| Target | Each mouse clikc becomes a gunshot |

### Fred Fish Disk 82

| | |
|---|---|
| Adventure | Port of the classic Crowther and Woods game |
| AmicTerm | V0.50 of a telecommunications program, with scripts, redial, beeps, enhanced file requester |
| D2D-Demo | Demo version of Disk-2-Disk from Central Coast Software |
| DX-Synth | Voice filer program for Yamaha DX series synthesizers, update to disk 38 |
| DiskMan | V1.0 of another DirUtil program |
| Icons | Miscellaneous new icons |
| Panl | Universal MIDI patch panel, v1.2 |
| Rocket | Another Workbench hack, plays Lunar Lander |
| Sand | Game of sands following your pointer. |

### Fred Fish Disk 83

This disk contains a demo version of TeX from N Squared. It is limited to small files, and the previewer can only display ten pages or less, and only a small number of fonts are provided.

### Fred Fish Disk 84

| | |
|---|---|
| AudioTools | Programs from Rob Peck's July/August Amiga World article |
| BitLab | Bitter experimentation program, V1.2, update to FF69 |
| Ed | Simple editor, similar to Unix 'ed', based on the editor in Software Tools. |
| GravityWars | Game of planets, ships and black holes, v1.04, update to disk 70. |
| HunkPad | Adds legal padding to executables for Xmodem transmission. |
| PipeHandler | An AmigaDOS pipe device which supports named pipes and taps. V1.2 |
| PopCLI | V3.0 of a hot-key to invoke a CLI window, with screen blanker, update to disk 40. |
| Requester | Update FF34, file requester similar to DPaint. |
| ScottDevice | V33.1 of a 'mount'able MicroForge SCSI driver. |
| Viacom | Another Schwab hack, makes TV-like static on screen.Parody |

### Fred Fish Disk 85

| | |
|---|---|
| Csh | V2.06 of Dillon's 'csh'-like shell |
| FileReq | Source to wildcard file requester |
| Hide | Hides expansion memory from programs |
| ImageTools | Shareware tools to manipulation IFF images |
| LowMem | ServerShared library to aid in low memory situations |
| Plot6 | A star plotting program with source. |
| RawIO | Example of setting raw mode on standard input |
| Rocket | Lunary Lander for Workbench, with source. |
| VMore | "more"-like text viewing utility, v1.0 SE |
| Vnews | Simple Unix news reader. |

### Fred Fish Disk 86

| | |
|---|---|
| AutoPoint | Auto-selects window under the mouse pointer, with screensaver. |
| ClickToFront | Double-clicks in window brings it to front, v1.1, S-E-D |
| Cmd | V3.0 of a tool to redirect printer output toa file. |
| FileIISG-Demo | Demo of Softwood File IIsg, a database manager with sound and graphics. |

### Fred Fish Disk 87

| | |
|---|---|
| AdvSys | Adventure system from Byte May 1987, v1.2 E-D |
| AutoIconOpen | Fools Workbench to open disk icons, V1.2 update to disk 73, S-E-D |
| Claz | Converts IFF files to PostScript, V2.0, SED |
| Commodi | tiesMackraz's Commodities Exchange, an exec library to manage input handler, v0.4 |
| Diff | Update to disk 75 of Unix-like 'diff', S-E-D |
| Dme | V1.27 of Dillon's text editor, update FF74,E-D |
| DropShadow | V2.0 of program that puts shadows on Workbench, S-E-D |
| Elib | Shared library example in Manx C. |
| ID-Handler | An AmigaDOS device handler generates unique identifiers, V1.0, S-E-D |
| Install | Alternate AmigaDOS 'install' programs,SED |
| MemWatch | Waits for low memory trashing, V2.0, SED |
| MovePointer | Moves pointer to given location, S-E-D |
| MoveWindow | Move window to given location, S-E-D |
| MunchingSq | Munching Squares hack, S-E-D |
| PalTest | Example shows test to see if this is a PAL machine, S-E-D |
| Sc | Generates random scenery, S-E-D |
| Tek4695 | Tek4695 printer driver |
| WBDualPF | Example of dual-playfield screen, update FF41, S-E-D |
| WarpText | Fast text rendering routines, S-E-D |
| Yaiff | rExample IFF reader, S-E-D |
| Zoo | A file archiver like "arc", v1.42A, E-D |

### Fred Fish Disk 88 (see Fred Fish 89)

FF Disk 88 has been removed due to copyright problems

### Fred Fish Disk 89 (replaces Fred Fish 80)

| | |
|---|---|
| DirMaster | Disk catalogue program, V1.0a, E-D |
| FuncKey | Shareware function key editor, V1.01, E-D |
| MFF-Demo | Demo of MicroFiche Filer database prog |
| ScreenShift | Adjust screen position ie Preferences,SED |
| Snake | Bouncing squiggly lines demo, S-E-D |
| AutoEnguirer | screen contraption requester improvement S-E-D |
| DemoLition | Display Hack S-E-D |

### Fred Fish Disk 90 (replaces Fred Fish 80)

| | |
|---|---|
| AmiGazer | Night sky viewer of 1573 stars, set date, time, day. E-D |
| CardFile | AmigaBasic card file study aid. E-D |
| Conman | Console handler replacement gives line editing and history to most progs, v0.98,ED |
| IMandelVroom | Slight update to disk 78 Mandelbrot program, E-D |
| NewDemos | Replacements for lines and boxes demos that take less CPU time, E-D |
| Othello | Game of Othello, E-D |
| PrinText | Displays text files with gadgets, speech, IFF display, v1.2, E-D |
| PrtDrvGen | Automatic printer driv. generator,v2.2b,ED |
| RainBench | Cycles colors of WB backdrop or text. ED |
| ShortCut | Makes single-key shortcuts for entering commonly typed CLI commands,&custom macros. E-D |
| ShowPrint | Displays and prints all sizes of IFF pictures & controls printer output styles, v2.0 E-D |
| Sizzlers | Graphics demos, v1.7.0, E-D |
| Timer | Small Workbench timer counts time and $/ minute, E-D |
| Tools | Inovatronics tools: a memory editor, memory disassembler, ASCII chart, and calculator. E |

### Fred Fish Disk 91

Adventure Definition Language (ADL) a superset of an older language called DDL by Michael Urban, Chris Kostanick, Michael Stein, Bruce Adler, and Warren Usui. ADL enhancements by Ross Cunniff. Included are sources to the ADL compiler, interpreter, and debugger. Binaries combined by Ross with Lattice 3.03. CLI enviornment only. Documentation is available from the authors.

### Fred Fish Disk 92

| | |
|---|---|
| As6502 | portable 6502 assembler, C source, by J. Van Ornum, Amiga port by Joel Swank |
| Bawk | Text processor update from FF65 Inspired by UNIX awk. Searches files for patterns, performs actions based on patterns. By Bob Brodt; Amiga port by Johan Widen |
| HunkPad | update of FF84 version. by J. Hamilton,pads an object file to a multiple of 128 bytes for better xmodem transfer. S E |
| Less | Like Unix "more", better, version 1.2 update of FF74. Scrolls Back and f orward. S E by Mark Nudelman, Amiga port by Bob Leivian. |
| Ndir | Library that implements the 4BSD unix dir access routines by MikeMeyer. S |
| Parse | Recursive descent expression parcer, computes, and prints expressions. includes transcendental function support. c Source included. by J Olsen |
| Shar | Two programs to pack and unpack shell archives includes C source. by Fabbian G. Dufoe |
| SmallLib | 8 times smaller Amiga.lib replacement, binary only. by Bryce Nesbitt |
| UUencode | Encode/decode binary files for e mail or text-only methods. Update of FF53, includes checksum technique, compatible with older versions, plus transparent to older versions options. By Mark Horton, modified by Alan Rosenthal and Bryce Nesbitt. |

### Fred Fish Disk 93

| | |
|---|---|
| Dme | Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, title-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code. by Matt Dillon |
| MicroEmacs | Version 3.8i, update to FF61 includes source. Orig by Dave Conroy multiple modifications by Daniel Lawrence |

### Fred Fish Disk 94

| | |
|---|---|
| AudioTools | Demo programs from Rob Peck's July/Augustt issue of AmigaWorld on accessing the audio device. V2,update of FF84. S. by Rob Peck |
| ClickUpFront | Similar in function to ClickToFront prog (FF86), bring windows tofront by clicking on any part of them. V 1.0. by Davide Cervone SE |
| HeliosMouse | Automatically activate a window simply by moving the mouse pointer into the window. V 1.0. Includes source. By Davide Cervone |
| IFF2Ps | Convert any IFF file to postscript for printing or viewing on a postscript compatible device. Version 1.2, by William Mason and Sam Paolucci E |
| ModulaTools | Various Modula 2 programming routines. by Jerry Mack |
| Terrain3d | Pseudo-random 3d relief scenery generator, update of "sc", FF87. by Chris Gray, 3d by Howard Hull |

### Fred Fish Disk 95

| | |
|---|---|
| Cmd | redirects the serial.device or parallel. device output to a file. Capture print jobs, debug or "offline" printing.V4 By C Scheppner SE |
| CygnusEdDemo | Demo of CygnusSoft's CygnusEd editor, a multiplefile, multiple feature editor.Includes demo 3.0 of MandFXP. by CygnusSoft Software E |
| Gomf | "Get Outa My Face" makes the Guru go away to allow clean-up & shutdown more cleanly. V 1.0, by Christian Johnsen E |
| Journal | records sequence of mouse & keyboard events, stored in a file for future playback. Good for demos or documtenting bugs. E. by D. Cervone |
| MergeMem | attempts merging of MemList entries of sequentially configured ram boards. When successful, allows allocating a section of memory which spans both boards. V 2, update of FF56. by Carolyn Scheppner SE |
| PrinterStealer | Asimilar to "Cmd", allows diversion of output destined for printer to a file. Binary only, Source avai. from authors. by A. Livshits & J-M Forgeas |
| Record-Replay | similar to "Journal", records and plays back mouse and keyboard events. B only. source avail. from authors, Alex Livshits & J-M Forgeas |

### Fred Fish Disk 96

| | |
|---|---|
| AnimPlayer | Animation reader and displayer by the combined efforts of Videoscape, Sculpt3D, Silver, Forms-In-Flight, and Animator:Appreticeby M Hashetal. |
| Chess | Usenet posted Amiga port, non-Amiga interface. High playability. V 1.0, S. by J. Stanback, Amiga port by B. Leivian |
| Hackbench | provides source for WB-Likeprog. for experimentation & validation of new interface ideas. Not a WB replacement. by Bill Kinnersley |
| Label | Print labels with arbitray text. V 1.3, Source available from author, M.Hansen |
| LineDrawer | Produces line drawings based on drawing commands stored in a text file. Includes demo that draws an outline map of the USA and state borders. V1.0, SE. by John Olsen |
| PopUpMenu | Example code implementing pop-up menus, reasonably compatible with Intuition menus.SE. by Derek Zahn |
| Tek4695 | Tektronix 4695/4696 printer driver. SE. by P Staub |
| TimeRam | Fast and Chip ram test prog,E by B Takahashi |
| WarpText | Fast text rendering routines, to be linked with application progs.Text display "as fast or faster than "blitz"". V2.0 update of FF87, S by Bill Kelly |

### Fred Fish Disk 97

Replaces FF57 for Copywrite problems

| | |
|---|---|
| CutAndPaste | Implementaions of Unix cut and paste commands. by John Weald |
| GraphIt | Progran to plot simple functions in 2 or 3 dimensions. by Flynn Fishman |
| Juggler | V1.2 of robot juggler animation. Uses HAM mode and ray tracing. by Eric Graham |
| MouseReader | Shareware program to read text files & view IFF files using only the mouse. by William Betz |
| Splines | Prog to demonstrate curve fitting & rendering techniques. by Helene (Lee) Taran |
| Shm | Simple graphics demo, approximately simulates the motion of two interacting pendulums. Includes Source by Chris Ediss |

### Fred Fish Disk 98

| | |
|---|---|
| Access | 16 color terminal program based on Comm V1.,34. Includes Macro window, custom gadgets, colorized menus, etc. V. Beta 0.18 by Keith Young ,comm by D.J.James. E. |
| Backup | Writes AmigaDos disks as the backup destination. recover files from the backup disk. Requires manual decisions on disk structure. by Alan Kent SE |
| DCDemo | DishCat 2.3, a disk catalog program, demo limited to cataloging 100 files at a time. by Ed Alford, MicroAce Software |
| HdDriver | WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED |
| QBase | Quick-Base, a "MailBase Management utility", define and maintain a maximum of 200 records per file. by Kevin Harrise E |
| Thai | Thai language quiz program. Speak or type english/Thai sentences from supplied file. by Alan Kent SE |

### Fred Fish Disk 99

A-Render Version .3 a Ray- Tracing Construction Set for the Amiga Computer by Brian Reed ED

### Fred Fish Disk 100

| | |
|---|---|
| Berserk | Must see animation, by Leo Schwab |
| Conman | Console handler replacement, provides line editing and command line histories transparent to application prog uses CON: windows. Shareware V1.0 by W Hawes. E. |
| WBLander | Workbench display hack game, upgrade of "Rocket" on FF85, now with sound effects. By Peter da Silva. E |

### Fred Fish Disk 101

| | |
|---|---|
| CirPlane | Circular plane generator for VideoScape3D. Generates a clockwise circular polygon with the specified number of vertibies. V1.0 by Thad Floryan SE |
| IconAssembler | Change Workbench Icons with IFF-brush files by Stefan Lindahl E |
| Microspell | Standalone spelling checker scans text files and reports errors. 1000 common word list, 43,000 word main dictionary with multiple user dictionary support. Interfaces with MicroEMACS 3.9 with an emacs macro to step through the source file, stopping at suspect words and allowing the user to option. V1.0 by Daniel Lawrence, SED |
| Midi | midi library and utility set. Includes Midi monitor, routing utility, status utility, and more. by Bill Barton SED |
| PsIntrp | Postscript Interpreter reads and previews files on screen. by Greg Lee S(assy)E |
| StartUps | Three C startup file replacements for standard Astartup.obj and LStarup.obj. Options include (1) BothStartup.obj, for the WorkBench programs or CLI programs with or without command line parameters. (2) WBStartup.obj, for WorkBench programs or CLI programs that require no command line parameters. (3) CLIStartup.obj for CLI programs that require command line parameters but do not need to be WorkBench runnable. by Bryce Nesbitt SE |

### Fred Fish Disk 102

| | |
|---|---|
| Dbug | Machine independent macro based C debugging package. Update FF41. by F Fish profiling support by Binayak Banerjee SE |
| Match-stuff | Heavy duty text pattern matching stuff. includes simple match text replacement capability. By Pete Goodeve |
| Sectorama | Recover lost or damaged data from floppy or hard disks or repair a damaged volume. by David Joiner E |
| SiliCon | Smart input line interpreter with window for full editing. Upgrade FF50 by P Goodeve., E |
| Xicon | Use icons to call up scripts containing CLI commands. V2.0 upgrade of FF31.by Pete Goodeve E |

### Fred Fish Disk 103

| | |
|---|---|
| AvlTrees | Library and test prog. implement routines for creating and using trees held in memory.S. |
| Calc | A programmable RPN calculator. |
| Cref | A C cross ref. prog. S. |
| DosKwik | A pair of progs. which allows you to save files to one or more floppies for quick loading. Doesn't store Dos format. |
| IntuiDos | A prog. to improve control and handling of the material on all disks in 'CLI-area' . |
| MFF-Update | A text import util. for MicroFiche Filer(demo on FF 89) and updates to some PD disk library databases. |
| Pack-It | Takes all files the files and dirs. on a disk & packs them into a single file. for modem. |
| Sol | Amiga version of solitaire. |

### Fred Fish Disk 104

| | |
|---|---|
| Analyticalc | Is a large and powerful spreadsheet prog. |

### Fred Fish Disk 105

| | |
|---|---|
| AsmProgs | Misc. assembly tools. Includes some S. |
| BasicProgs | LeastSquare solves least square probs & graphs results. S. |
| Bison | A replacement for unix "yacc" command. S. |
| Dmouse | Another prog in the tradition of display hacks". S. |
| FlamKey | Allows keyboard and mouse inputs to be locked until a password is entered. |
| GravityWars | Game of planets,ships and black holes, v2.0 update to FF84. |
| IPo2C | A util. to write a C-lang definition to mimic the intuition pointer.S |
| Pere-et-Fil | Ex. of creating & using reentrant processes. S. |
| Record Replay | Similiar to "Journal" v2.0 update to FF95. |

### Fred Fish Disk 106

| | |
|---|---|
| Funckey | Shareware function key editor, v1.1update to FF89. Source avail. from author(Anson Mah). |
| MoreArt | A small selection of some Amiga artwork. |
| QuickFlix | An IFF slideshow and cel animation prog,v0.13. |
| RistNolla | A Finnish game. Also called Go-Moku. v1.0 |

### Fred Fish Disk 107

| | |
|---|---|
| Csh | V2.07 of Matt Dillon's csh like shell.S. |
| Diff | A util.,similiar to other common "diff" programs.S. |
| ProSuite | Suite provides ex. code of facilities such as FileIO Requester, XText, DoRequest. & tutorial on how to program the Amiga. Book 1.01.S |
| SVTools | Some useful tools. S. |

### Fred Fish Disk 108

| | |
|---|---|
| AList | Dir listing prog. based on LD4 prg S |
| DirMaster | Disk cataloger. v1.0b, update to FF89. S. |
| Dots-Perfect | Printer Driver for an Epson MX80 printer with upgrade kit installed. S. |
| MonIDCMP | Lets you monitor the IntuiMessages that pass through an IDCMP window. Prints the message class,mouse coordinates,qualifier values. Great for debugging. S. |
| PrintPop | A util. to send common control settings to the PRT: device. S. |
| Sectorama | Utilities to recover lost or damaged data from floppies & hard disks. v1.1, an update toFF102. |
| Tek | Vt100 emulator for a Tektronix 4010/4014. (V2.6) update to FF52. S. |
| Zoo | File archiver, like "arc". v1.24B. update to FF87 |

### Fred Fish Disk 109

| | |
|---|---|
| Machine | A new animation. |
| SimCPM | A CP/M sim,simulates 8080 along with h19 emulation. S. |
| UUpc | Hook up your Amiga as a usenet node. S. |

### Fred Fish Disk 110

| | |
|---|---|
| A68k | A 68000 assembler written in C. S. |
| Pdc | An optimizing C compiler for the 68000 processor. update to FF53. but not based on the code of that disk. |

### Fred Fish Disk 111

| | |
|---|---|
| AmyLoad | A graphical monitor of cpu, blitter, & memory use. Includes two components; load.device,monitors system parameters, & amyload, which is the user interface & display program. by Jeff Kelley SE |
| AssignDev | Assigns multiple names to a given device. modified version of the original released on disk number 79. By Phillip Lindsay, mod by Olaf Seibert SE |
| Gauge | Continuously displays memory usage in a vertical bar graph Binary only. By:Peter da Silva |
| HeliosMouse | Another "sunmouse"prog. Automatically activates a window by mouse pointer V 1.1, update to disk 94. By:Davide Cervone SE |
| Labels | Alphabetic & numeric ordered cross reference lists of defined system constants. Recommended for debugging purposes only, use the symbolic values in progs! By:Olaf Seibert |
| Mandel | Another mandelbrot generator program, with bits & pieces of code from C. Heath & R.J. Mical. By:Olaf Seibert S |
| PopLife | A PopCLI type that plays life all over your screen. Lots of bits & pieces from Tomas Rokicki's blitlab & John Toebes' PopCLI. By:Olaf Seibert S |

### Fred Fish Disk 112

| | |
|---|---|
| BeachBirds | Beach scene portrayed by sprites & sound 512K machine. By:Jerrold Tunnell B only. |
| Bully | Pushes all open screens around (thus the name "bully"). Show more than one demo at a time By:Mike Meyer S |
| DropShadow | Dropshadow V2.0, use with Bryce Nesbitt's Wavebench demo. B only. By:Jim Mackraz |
| HagenDemos | "RGB" & "Focus". RGB requires one meg, B only. By:Joel Hagen |
| Viacom | Latest version of viacom for use in conjuction with WaveBench demo. B only. By:Leo Schwab & Bryce Nesbitt |
| WaveBench | A neat screen hack, & runs on 512K machines. For more laughs, try in conjunction with Viacom or Ds (Dropshadow). Includes S. By:Bryce Nesbitt |

### Fred Fish Disk 113

| | |
|---|---|
| AmiCron | Simple Unix "cron" type program,a background task uses a disk-resident table to automatically run certain tasks on a regular basis, at specific times. V 2.3, includes S. By:Steve Sampson, Amiga port by Rick Schaeffer |
| Dme | V 1.28f of Matt's text editor. A simple WYSIWYG editor designed for programmers. It is not a WYSIWYG word processor Features include arbitrary key mapping, fast scrolling, title-line statistics multiple windows, & ability to iconify windows. Update to FF93, includes S. By:Matt Dillon |
| DosDev Example | DOS device driver in Manx C. Version 1.10, includes S. By:Matt Dillon |
| M2Amiga | Demo of the final product M2Amiga. A fast single pass Modula-2 compiler with editor, linker, a small set of interface & standard libraries. Compiles only small demo programs by limiting codesize & imports. Further development of the ETHZ compiler on Disk 24. B only. Demos with Source. By:R. Degen, C. Nieder, M. Schaub, J. Straube (AMSoft) |
| NoIconPos | Clears position info of any icons, allows WorkBench to pick a new place for the icon. Useful for disk & drawer icons where Snapshot rewrites the icon & the window information. Modula-2, another demo for M2Amiga By:Markus. Schaub |

### Fred Fish Disk 114

| | |
|---|---|
| CDecl | English to C (and vice versa) translator for C declarations, a must for anyone except possibly the most hardcore C guru. By:Graham Ross, S |
| Vt100 | V2.7 of vt100 terminal emulator with kermit & xmodem file transfer. Includes a few bug fixes posted to Usenet shortly after the posting of v2.7. Update to FF55. Includes S. By:Dave Wecker |
| WBLander | a special version of the WBLander program from FF100. Ending is unique. Effective use of sound, Includes S. By:Peter da Silva & Karl Lehenbauer |

### Fred Fish Disk 115

| | |
|---|---|
| Killer | Masterful Video commercial of the Amiga, Beatles music, requires one meg of memory to run. Binary only. By:Robert Wilt |
| Marketroid | Another devious sprite oriented demo with lots of "in" jokes. 512K required, includes S. By:Leo Schwab |

### Fred Fish Disk 116

| | |
|---|---|
| Movies | A ram animation system with three different example animations; Kahnankar Rocker, & F-15. Kahnankas & Rocker run on a 512K Amiga & show off overscan HAM mode. Includes a animation player program (movie), animation builder programs (dilbm, pilbm), & a text/graphics display program (vilbm). By:Eric Graham & Ken Offer |

### Fred Fish Disk 117

| | |
|---|---|
| AMUC_Demo | A really neat horizontal scrolling demo that is a 2400 x 200 pixel 32 color IFF picture composed of digiview snapshots of members of the AMiga Users of Calgary, superimposed on a very wide picture of the Calgary Skyline. B only. By:Stephen Vermeulen & Stephen Jeans |
| ExP_Demo | Demo version of Express Paint 1.1., used to create the scrolling demo picture in the AMUC_Demo drawer on thi disk. B only. By:Stephen Vermeulen |

### Fred Fish Disk 118

| | |
|---|---|
| Empire | This is a complete rewrite, from the ground up, in Draco, of Peter Langston's Empire game. A multiplayer game of exploration, economics, war, etc, can last months.Played either by local keyboard or through modem.V1.0, shareware, & includes S code. By:Chris Gray, original game by Peter Langston |
| HAMmmm | Displays lines whose end points are bouncing around the screen, which is a double buffered HAM screen. The Y positions of the points are continuously copied into an audio waveform that is played on all four channels, & the pitch of a just intoned chord is derived from the average X position of these points. JForth. Source By:Phil Burk |
| Stars | Based on original code by Leo Schwab, has credits longer than the actual demo. Runs on 512K Amiga. B only. By:Hobie Orris |
| WireDemo | Demonstrates the Amiga's line drawing speed. Runs on a 512K Amiga. Includes S. By:Matt Dillon |

### Fred Fish Disk 119

| | |
|---|---|
| MicroEMACS | Version 3.9e of Daniel Lawrence's variant of Dave Conroy's microemacs. This is an update to the version released on disk 93. Also included, for the first time, is extensive documentation in machine readable form. Includes source. Author: Dave Conroy, MANY enhancements by Daniel Lawrence |

### Fred Fish Disk 120

| | |
|---|---|
| Amoeba | This clone of Space Invaders is one of the best freely redistributable games for the Amiga to date. Unlike many commercial games, it even works correcty in a multitasking environment (by not requiring you to reboot just to play a game). Highly recommended! Binary only. Author: LateNight Developments |
| BackGammon | A graphical Backgammon game done as an undergraduate A.I. course project. Version 1.0, includes source. Author: Robert Pfister |
| Bankn | A complete checkbook system offered by the author as shareware. Version 1.3, binary only. Author: Hal Carter |
| EgyptianRun | Cute little "road race + hazards" type game. Version 1.1, binary only, shareware, source available from author. Author: Chris Hames |
| IconImage | Program to replace an old icon image with a new image, without affecting icontype, drawer data, etc. Includes source. Author: Denis Green |

### Fred Fish Disk 121

| | |
|---|---|
| BasicStrip | An AmigaBASIC program that helps to convert programs written in other forms of Basic to AmigaBASIC. Author: George Trepal |
| DataPlot | A shareware plotting program written in AmigaBASIC. Also includes a least squares curve fit program. Author: Dale Holt |
| Plot | A shareware 3-D graphing program written in AmigaBASIC, with some sample output plots. Source available from author. Author: George Trepal |
| Stairs | This AmigaBASIC program demonstrates a musical illusion based upon perceptual circularity of widely spaced tones whose volumes are defined as a sinusoidal relationship to their frequency. Author: Gary Cuba |
| Uedit | Version 2.3 of this nice shareware editor. Has learn mode, a command language, menu customization, and other user configurability and customizability features. Binary only, shareware, update to version on disk 60. Author: Rick Stiles |

For PDS orders, Please use form on page 112

| | |
|---|---|
| WBColors | A simple little program to change the Workbench colors to a predetermined color set, for programs that expect to be booted off their distribution disk but instead are run from a hard disk. Includes source. Author: Stefan Lindahl |

**Fred Fish Disk 122**

| | |
|---|---|
| Asteriods | Nicely done "blast the asteriods" type game. Unique feature is that all the images and sounds are replaceable by the end user. So instead of ships and rocks, you can have an Amiga against a horde of IBMPC's if you wish. Author: Rico Mariani |
| Iff2Pcs | An interactive puzzle program that takes any IFF file containing up to 16 colors, and breaks it up into squares to make a puzzle which the user can then piece back together again. Version 1.0, includes source. Author: Ali Ozer |
| Names | A shareware program to create and manage mailing lists. Binary only. Author: Ernie Nelson |
| Pr | A little utility to print listings in different formats. Similar to the Unix "pr" program. Includes source. Author: Samuel Paolucci |
| PushOver | A neat little board strategy game, in AmigaBASIC. Push your pieces onto the board until you get five in a row in any direction. Includes source. Author: Russ Yost |
| PuzzlePro | Create a puzzle from an IFF picture, which the user can then piece back together again. Written in AmigaBASIC. Version 1.0, binary only, shareware, source available from author. Author: Syd Bolton |

**Fred Fish Disk 123**

| | |
|---|---|
| Arp | ARP stands for "AmigaDOS Replacement Project". Arp is an effort led by Charlie Heath of Microsmiths Inc., to replace the current DOS in a compatible fashion, so that current programs will continue to work. Arp also makes whatever improvements are possible, so that current and future programs will work better. Author: Various authors contributed work |
| Car | This animation is one of Allen's entries to the Badge Killer Demo Contest. It apparently is an inside joke relating to a well known Amigan's experience with a certain highend graphics hardware manufacturer. Author: Allen Hastings |

**Fred Fish Disk 124**

| | |
|---|---|
| Icons | Some sample animated icons. You might find just the icon for that refugee CLI program you've been meaning to make runable from the WorkBench environment. Author: L. Pfost |
| Tarot | An AmigaBASIC program written by the author as an exercise for learning BASIC. Contains some nice graphic renditions of tarot cards. Author: L.Pfost |

**Fred Fish Disk 125**

| | |
|---|---|
| ElGato | This animation is Kevin's entry to the Badge Killer Demo Contest. It also has a background music arrangement, that requires Sonix to use. Author: Kevin Sullivans |

**Fred Fish Disk 126**

| | |
|---|---|
| Colour | A program to manipulate the colors of specific named screens, saving their current color sets to data files, loading new color sets from data files, or interactively changing the colors. Includes source. Author: John Russell |
| Dance | These two programs, "dancing polygons", are John's entry to the Badge Killer Demo Contest. They are variations of one another, but demonstrate the range of colors available on the Amiga. Includes source. Author: John Olsen |
| HBHill | This animation is one of Kevin's entries to the Badge Killer Demo Contest. It is the first known animation that makes use of the Amiga's "Extra Half Brite" mode. Author: Kevin Sullivan |
| Iconify | A subroutine that creates an icon on the Amiga screen that can be subsequently dragged around, and double-clicked on. You can use this to have your programs "iconify" themselves to temporarily get out of the user's way. Includes source and demo program. Author: Leo Schwab |
| OnlyAmiga | This animation is Iqbal's entry to the Badge Killer Demo Contest. It consists of tree balls being juggled by pyramids rotating on their tops. Author: Iqbal Singh Hans |
| Suplib | The support library needed to rebuild various programs of Matt's from the source, including DME, DTERM, etc. Includes source. Author: Matt Dillon |
| VCheck | Version 1.2 of the virus detection program from Commodore Amiga Technical Support. This version will test for the presence of a virus in memory, or on specific disks. Binary only. Author: Bill Koester. |

**Fred Fish Disk 127**

| | |
|---|---|
| Bounce | This program is Steve and Tom's entry for the Badge Killer Demo Contest. It creates little dots that bounce around and multiply. Includes source. Author: Steve Hansel and Tom Hansel |
| Nemesis | This demo is Mark's entry to the Badge Killer Demo Contest. It is quite small for what it does, and won fifth place in the contest. Binary only. Author: Mark Riley |
| Ripples | This animation is one of Allen Hastings' entries to the Badge Killer Demo Contest. Unlike most other animations, it shows a fixed object from a moving point of view, rather than a moving object from a fixed point of view. Author: Allen Hastings |

**Fred Fish Disk 128**

| | |
|---|---|
| Dis | A 68000 disassembler, written in 68000 assembler. Includes source. Author: Greg Lee |
| DropCloth | lets you place a pattern, a 2 bitplane IFF image or a combination of a pattern and image, into the WorkBench backdrop. Version 2.2, shareware, binary only. Author: Eric Lavitsky |
| LedClock | An extremely simple clock program, for interlaced screens only. Includes source. Author: Ali Ozer |
| MRBackUp | A hard disk backup utility, that does a file by file copy to standard AmigaDOS floppy disks. Includes an intuition interface and file compression. Version 1.3, includes source. Author: Mark |
| RinfretPaint | A simple screen painting program, written in web. Requires web preprocessing program to rebuild from source. Includes source in web. Author: Greg Lee |
| PrtDriver | A printer driver for the Toshiba "3 in one" printer in its Qume (best) mode. Includes source in C and assembler. Author: Rico Mariani |
| SDBackUp | A hard disk backup utility. CLI interface only. Does file compression. Version 1.1, binary only. Author: Steve Drew |
| Sed | A clone of the Unix sed (Stream EDitor) program. Includes source. Author: Eric Raymondw |
| Keys | A "hot-keys" program that binds keyboard function keys to window manipulation functions (window activation, front to back, moving screens, etc). Includes source. Author: Davide Cervone |

To Be Continued......

## In Conclusion

To the best of our knowledge, the materials in this library are freely distributable. This means they were either publicly posted and placed in the Public Domain by their Author, or they have restrictions published in their files to which we have adhered. If you become aware of any violation of the author's wishes, please contact us by mail.

## IMPORTANT NOTICE!

This list is compiled and published as a service to the Commodore Amiga community for informational purposes only. Its use is restricted to non-commercial groups only! Any duplication for commercial purposes is strictly forbidden. As a part of Amazing Computing™, this list is inherently copyrighted. Any infringement on this proprietery copyright without expressed written permission of the publishers will incur the full force of legal actions.

Any non-commercial Amiga user group wishing to duplicate this list should contact PiM Publications, P.O.Box 869, Fall River, MA 02722. PiM Publications is extremely interested in helping any non-commercial support for the Amiga.

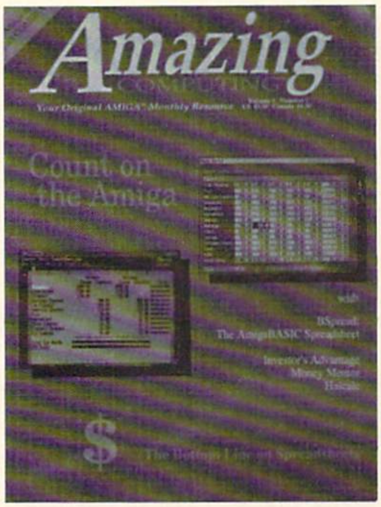
•AC•

**To Order Public Domain Software,**
please use the form on page 112.

# Amazing COMPUTING

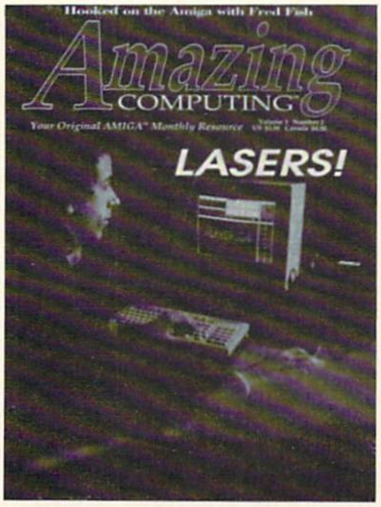
## Back Issues

(continued from page 49)


Amazing COMPUTING
Your Original AMIGA Monthly Resource
Count on the Amiga
BSpread: The AmigaBASIC Spreadsheet
Investor's Advantage
Money Mentor
Halcalc
The Bottom Line on Spreadsheets

Volume 3 Number 1 1988

**AmigaNotes** by Richard Rae
Digital music generation on tha Amiga.
**C Animation Part IV** by Michael Swinger
Just when you thought it was safe to go back in the C waters...
**Forth** by John Bryan
Sorting out CHIP and FAST memory on the Amiga.
**The Big Picture** by Warren Ring
Daring assembler language programming; CLI system calls and manipulating disk files.
**Bug Bytes** by John Steiner
**Roomers** by The Bandito
Amiga Dos1.3? 80386-based BridgeBoard for the A2000? More!
**As I See It** by Eddie Churchill
Opinions, obsevations, & the birth of a new software generation.
**68000 ASsembly Languaage Programming** by Chris Martin
"Create a multi-color screen without using Intuition routines!"
**Modula-2 Programming** by Steve Faiwiszewski
A new contender bursts on to the modula-2 scene!
**The Amicus Network** by John Foust
Public domain update, Commodore politics, and developer info.
**Amicus Network Special Report: Fall COMDEX** by J. Foust
Commodore at COMDEX and new products
**The ultimate Video Accessory: Part II** by Larry White
**Life: Part II** by Gerald Hull
"A detailed look at efficient use of the Amiga blitter."
**FormatMaster: Professional Disk Formatting Engine**
by Carl Mann
Put Batch language to work on the drudgery of disk formatting.
**BSpread** by Brian Catley
A full featured AmigaBASIC spreadsheet you can program!
**AmigaForum Transcript ed.** by Richard Rae
Zoom in on Commodore/ Amiga's Dave Haynie.
**Halcalc Review** by Chuck Raudonis
"A straightforward, easy to use, functional spreadsheet."
**VIP Professional Review** by Suzanne Mitchell
Easy stock porfolio management on the Amiga.
**Money Mentor Review** by Stephen Kemp
A personal finance system beyond your checkbook.
**Investor's Advantage Review** by Richard Knepper
plus "Poor Man's Guide to the Stock Market."


Hooked on the Amiga with Fred Fish
Amazing COMPUTING
Your Original AMIGA Monthly Resource
LASERS!

Volume 3 Number 2 1988

**Laser Light Shows with the Amiga** by Patrick Murphy
Lasers and the Amiga: A Dazzling Tandem
**The Ultimate Video Accessory: PartIII** by Larry White
Take the final stepstoward designing your own videos.
**Our First Desktop Video** by Larry White
A step-by -step guide to organizing and presenting your first Amiga video.
**Hooked on the Amiga with Fred Fish** by Ed Berkovitz
Inside views from the man behind all those "Fish" disks.
**Photo Quality Reproduction with the Amiga and Digi-View**
by Stephen Lebans
Stunning Digi-View images look great in hardcopy, too!
**Balancing your Checkbook with WordPerfect Macros**
by Steve Hull
Hand your checkbook worries over to the Amiga.
**More Basic Text** by Bryan Catley
Displaying text on the screen can be even easier!
**Life: Part III** by Gerald Hull
The sries winds up with the famed nine-blit calculation and source to LIFER.
**Solutions to Linear Algebra through Matrix Computations**
by Ropbert Ellis
Simplify matrix algebra with basic operations & routines.
**The Amicus Network** by John Foust
A scrapbook look at the facews behind the Amiga.
**Roomers** by The bandito
Amiga 3000, Virus news, and "Will the real Laser Toaster Please stand up?"
**Bug Bytes** by John Steiner
**Modula-2 Programming** by Steve Faiwiszewski
Catching up with Calc-a source follow-up.
**68000 Assembler Language Programming** by Chris Martin
Graphics- Part II of Assemgram.
**Arazok's Tomb** by Kenneth E. Schaefer
"A terrifyig adventure into the world of the occult."
**AIRT** by Steve Faiwiszewski
An innovative icon-basede programming language.
**Forms In Flight** by Steve Pietrowicz
Render and Animate objects in 3D!
**Silicon Dreams and the Jewel of Darkness** by K E. Schaefer
Hang on every word in these classic text adventures.
**Leisure suit Larry** by Kenneth E. Schaefer
Tha ultimate Nerd spends one racy night in the fast lane.
**Two New Entries From Microbiotics** by John Foust
M501 Expansion & Starboard II MultiFunctionboard.
**Mindlight 7 and People Meter** by John Foust
Meet two Amiga products.
**Phantasie** by Kenneth E. Schaefer
Up Your chances in this "monster mash" with the *Amazing Phantasie Character Editor.*

•AC•

To order Back Issues,
Please use the order form on page 112

# Amaze Me

Please use this order form when subscribing to Amazing Computing™, ordering Back issues, or ordering Amiga™ Public Domain Software

Name ______________________________
Street ______________________________
City ____________________ St.____ Zip __________
Amount Enclosed ______________

**Please circle the appropriate item:** New Subscription Renewal

**Please start my subscription to Amazing Computing™ with the next available issue or renew my current subscription.** I have enclosed $24.00 for 12 issues in the U.S. ($30.00 Canada and Mexico, $35.00 overseas).
All funds must be in U.S. Currency on a U.S. Bank

**Back Issues: $4.00 each**
**(foreign orders add $1.00 each for Postage and Handling)**

**Please circle your Back issue choices below:**

Vol1.1 Vol1.2 Vol1.3 Vol1.4 Vol1.5 Vol1.6 Vol1.7 Vol1.8 Vol1.9 Vol2.1 Vol2.2 Vol2.3 Vol2.4 Vol2.5
Vol2.6 Vol2.7 Vol2.8 Vol2.9 Vol2.10 Vol2.11 Vol2.12 Vol3.1 Vol3.2

Public Domain Software:
$6.00 each for subscribers (yes, even the new ones!)
$7.00 each for non subscribers

**Please circle your Public Domain Software choices below:**

**Amicus:**

A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A11 A12 A13
A14 A15 A16 A17 A18 A19 A20 A21 A22 A23 A24 A25 A26

**Fred Fish:**

FF1 FF2 FF3 FF4 FF5 FF6 FF7 FF8 FF9 FF10 FF11 FF12 FF13 FF14
FF15 FF16 FF17 FF18 FF19 FF20 FF21 FF22 FF23 FF24 FF25 FF26 FF27 FF28
FF29 FF30 FF31 FF32 FF33 FF34 FF35 FF36 FF37 FF38 FF39 FF40 FF41 FF42
FF43 FF44 FF45 FF46 FF47 FF48 FF49 FF50 FF51 FF52 FF53 FF54 FF55 FF56
FFNA FF58 FF59 FF60 FF61 FF62 FF63 FF64 FF65 FF66 FF67 FF68 FF69 FF70
FF71 FF72 FF73 FF74 FF75 FF76 FF77 FF78 FF79 FFNA FF81 FF82 FF83 FF84
FF85 FF86 FF87 FFNA FF89 FF90 FF91 FF92 FF93 FF94 FF95 FF96 FF97 FF98
FF99 FF100 FF101 FF102 FF103 FF104 FF105 FF106 FF107 FF108 FF109 FF110 FF111 FF112
FF113 FF114 FF115 FF116 FF117 FF118 FF119 FF120 FF121 FF122 FF123 FF124 FF125 FF126

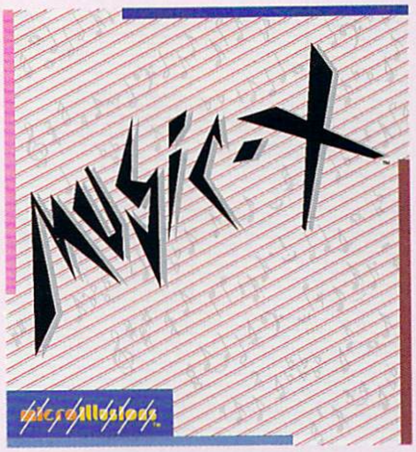
(NA denotes disks removed from the collection)

Please complete this form and mail with check or money order to:

PiM Publications, Inc.
P.O.Box 869
Fall River, MA 02722

Please allow 4 to 6 weeks for delivery

# MUSIC-X™



**MUSIC X SOFTWARE,** *like a fine instrument is crafted from the heart. It is more than an excellent tool, it is also a work of art.*

**COMMITMENT:** *We have committed ourselves to pushing ahead state of the art in professional music software, enabling you to open new worlds of creativity at a cost, both in hardware and software, that is well within the budget of any serious musician.*

**NO COMPROMISES** *or shortcuts have been tolerated as we designed this product. The master clock is accurate to 1 millisecond with a resolution of 192 clocks per quarter note. Sequences and library data can be any length, limited only by available memory — if you want, you can dump a 100K or larger sample into a library entry!*

**KEYBOARD MAPPING** *features allow almost any function of the sequencer to be controlled from a MIDI keyboard, footpedal, or other MIDI device. This includes starting/stopping the sequencer, initiating sequences, and even changing the key map itself!*

**REAL TIME:** *The system supports real-time recording of systems exclusive data, as well as full graphic-oriented and event-oriented editing of sequences. You can even record while in edit mode and watch notes appear on your edit display as you play them!*

**LIBRARIAN:** *A configurable librarian is included with the program. You can teach the librarian how to communicate with any MIDI instrument which outputs system exclusive data.*

**EDITING:** *An impressive battery of editing features will be supported. In fact, new editing features are being added daily as we interact with our network of working, professional musicians whose input has greatly contributed to the quality of this program.*

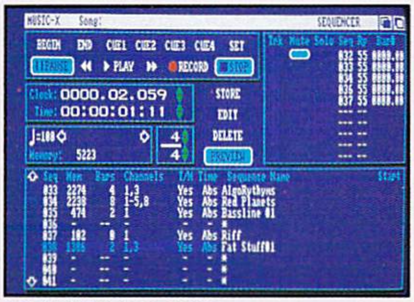
**COMMITMENT:** *Our commitment to music production does not stop here. A future product, Patch Editor Construction Kit, will allow you to create graphical patch editors for virtually any synthesizer you may own. Some technical knowledge will be required, but since patch editors, once created, can be traded between users, you should have no problem getting an editor for your needs.*

**THE POWER:** *Part of the power of Music-X comes from the computer it was created for: The Amiga, one of the most powerful and inexpensive personal computers available. At last you can run these many powerful applications in an environment that is a pleasure rather than a chore to use!*

**MICRO MIDI:** *Although Music-X will work with any of the many MIDI interfaces for the Amiga, we offer our own MIDI interface which we feel is a cut above. It features six outputs (each output switchable as OUT, THRU or OFF), two switch-selectable inputs, a channel loading indicator, and an external clock output (sync/start stop) for synchronizing older, non-MIDI drum machines, and a serial pass-thru!*

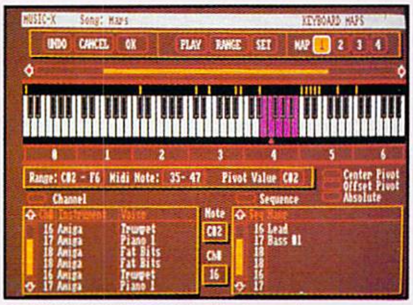
**MICRO SMPTE:** *This complete SMPTE Reader will allow Music-X to synchronize with video or audio tape decks. It connects to the Amiga parallel interface and includes a pass-thru so as not to interfere with printer operation. Our Micro SMPTE is compatible with all Amiga models (A500/A1000/A2000).*

**PHOTON VIDEO:** *Photon Video is a complete, integrated video animation system. It includes facilities for both 2-D and 3-D animation, as well as automatic tape transport control and real time playback of rendered images. Our 3-D rendering module supports variable light sources, shadows, transparency, and reflections in a 3-D environment. Other modules include Cel Animator, Object Editor and Transport Controller with SMPTE support.*



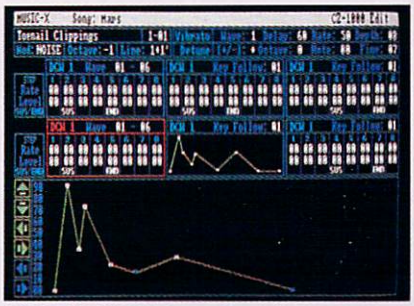
Clock: 0000.02.059
Time: 00:00:01:11
Memory: 5223

**SEQUENCER PAGE:** *Tape transport-type controls allow manipulation of up to 250 sequences; each contain 16 MIDI channels worth of data.*



Range: C02 - F6 Midi Note: 35- 47 Pivot Value C02

**KEYMAP EDITOR PAGE:** *Create keymaps by dragging the mouse over a selected area of the keyboard. The highlighted region can then be redefined in terms of real-time behavior.*



**PATCH EDITOR:** *A sample patch editor (CZ-1000) of the type that will be included with the product.*

COMMODORE
AMIGA™

microillusions™

*17408 Chatsworth St., Granada Hills, CA 91344*
*Inside CA 818/360-3715 •*
*Outside CA 800/522-2041*
*FAX 818/360-1464*

*Commodore is a registered trademark of Commodore Electronics, Ltd., Amiga is a registered trademark of Commodore-Amiga, Inc., and the Commodore-Amiga logo is a trademark of Commodore-Amiga, Inc.*

# DIGI VIEW

## DELIVERS ULTIMATE GRAPHICS POWER

Bring the world into your Amiga with Digi-View, the 4096 color video digitizer. In seconds you can capture any photograph or object your video camera can see in full color and with clarity never before available on a home computer. Digi-View's advanced features include:

- Dithering routines give up to 100,000 apparent colors on screen
- NewTek's exclusive Enhanced Hold-and-Modify mode allows for exceptionally detailed images
- Digitize images in any number of colors from 2 to 4096
- Print, animate, transmit, store, or manipulate images with available IFF compatible programs
- Digitize in all Amiga resolution modes (320x200, 320x400, 640x200, 640x400)

**"Digi-View sets new standards for graphics hardware"-*InfoWorld***

Digi-View is available now at your local Amiga dealer or call:
**1-800-843-8934**

**ONLY $199.95**

**NEWTEK**
INCORPORATED